REFERENCE MANUAL

FOR THE

DDP-24

GENERAL PURPOSE COMPUTER

COMPUTER CONTROL COMPANY, INC.

Copyright, 1963

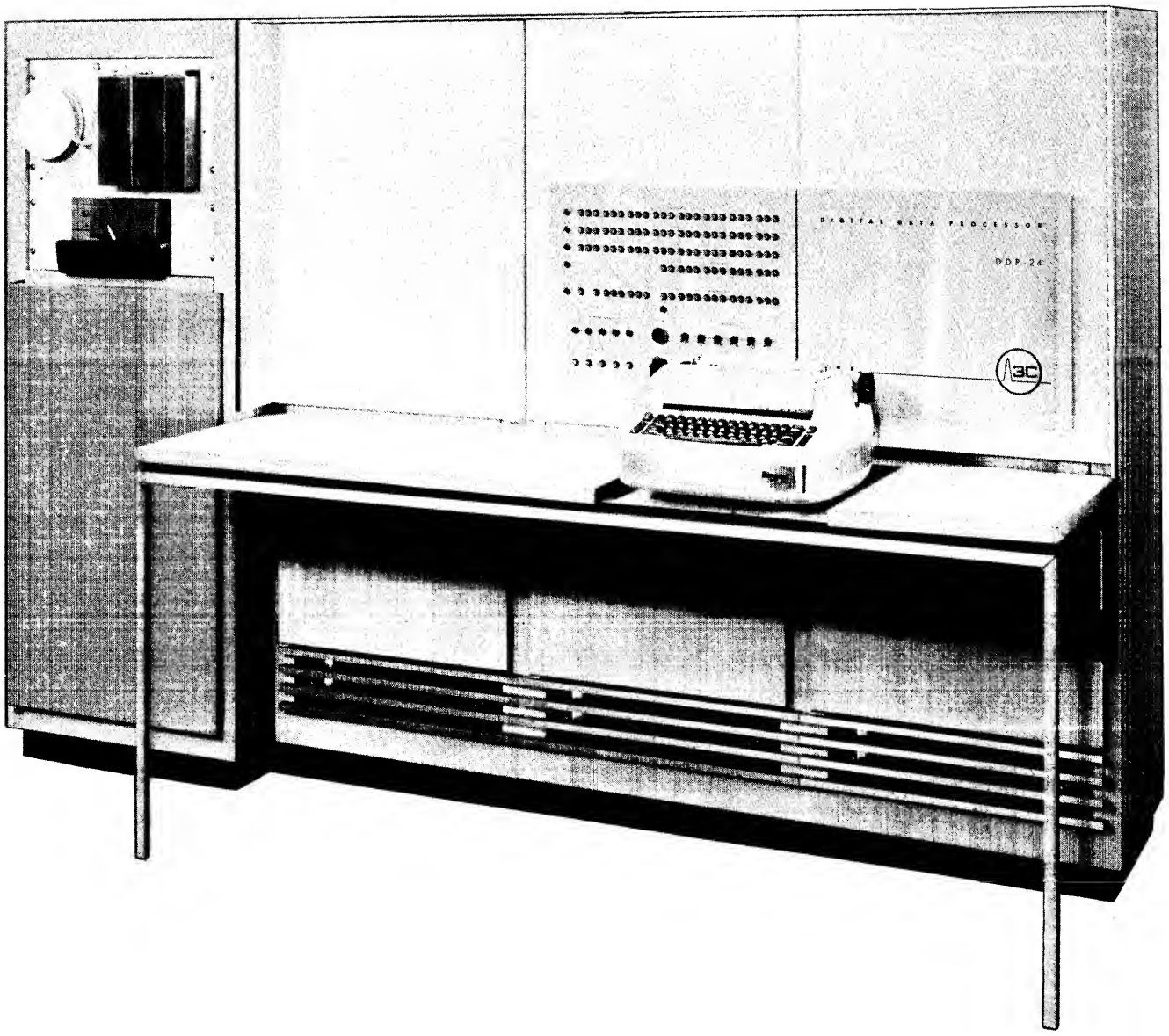


Figure 1 — Standard DDP-24

CONTENTS

GENERAL DESCRIPTION

INTRODUCTION

Computer Control Company's DDP-24 is a high-speed, solid state general purpose digital computer for real-time, on-line and scientific data processing. The DDP-24 design has evolved from the long experience of 3C in the manufacture of digital systems. This experience indicated that a powerful general purpose data processor with a wide range of input-output options would meet many user requirements.

The DDP-24 is a completely transistorized computer with 24-bit word length and sign magnitude code which is particularly convenient to represent negative numbers. The information representation gives maximum efficiency and convenience to the user. The DDP-24 has an expandable magnetic core memory which stores both program instructions and data which are part of or result from computer operations.

Parallel logic, fast memory cycle, and a powerful instruction repertoire make high speed operation possible. The processor uses binary logic. The command word format is single address, single operation with index and indirect addressing flags.

DDP-24 capabilities are easily expanded; options permit easy adaption of the general purpose device to special systems requirements at minimum cost. Typical options include additional input-output channels and interrupt lines, priority interrupt and greatly increased memory capacity. A full range of peripheral equipment can be connected to the computer.

The DDP-24 uses 3C's standard one megacycle S-PACS. These are general purpose circuit modules of proven reliability; the wide performance margins and the conservative circuit ratings account for the high reliability of the computer. (Further details are given in the 3C S-PAC manual.)

Standard input-output capabilities include four data input-output channels, a buffered character input channel, a buffered character output channel, and parallel 24-bit input and output channels. A variety of input-output devices may be used. Input-output modes of operation include automatic interrupt for any channel desired; several input-output devices may be operting simultaneously with the system. In addition, sixteen single line sense inputs and eight output control lines are provided.

A full range of peripheral devices are available for the DDP-24. They include high and low speed magnetic tape units, medium and high speed line printers, adapters for IBM card input and output units, XY plotters, paper tape spooler, and remote input-output consoles. Other peripheral devices can be easily connected such as cathode ray tubes, disc memories, drum memories, dataphone, analog-to-digital and digital-to-analog converters, multiplexers and distributors, teletype, and special instrumentation.

A full support program is provided including programmer and maintenance training, installation services, programming software, lease policy, logistic support, and an inventory of spare parts.

SPECIFICATIONS

Type:
Parallel binary, solid state

Addressing:
Single address with indexing and indirect addressing; flip-flop register

Word Length:
24 bits

Machine Code:
Sign-magnitude (refer to Appendix A)

Memory Type:
Coincident current ferrite core; non-volatile storage

Memory Size:
4,096 words, expandable to 32,768 words, all directly addressable

Memory Cycle Time:
5 microseconds

Memory Access Time:
3 microseconds

Speed:
Addition — 10 μsec
Multiplication — 31 μsec
I/O Word Transfer —5 μsec
I/O block transfer — 166,000 words per second

Standard Peripheral Equipment:
300 cps photoelectric paper-tape reader, with continuous and pulsed modes of operation; 60 cps paper tape punch; 15 cps electrical typewriter for input and output.
Off-line paper-tape preparation and printout.

Input-Output Channels:
Standard character buffer for both input and output, with parity, up to 6 bits. Standard parallel 24-bit input channel. Standard parallel bit output channel. Additional character buffers and parallel channels with or without flip-flop register are optional. Word forming buffers, direct memory access channels and fully buffered channels can also be provided.

Input-Output Modes:
Program controlled input-output (Ready Mode)
Automatic Interrupt
(Input-output channels can be connected to operate in either the Ready or Automatic Interrupt Mode)

Block Transfer
The Fill Memory Block command (FMB) and Dump Memory Block command (DMB)
Single line sense inputs and output control signals.
Direct memory access channel option.
Fully buffered channel option.
Priority Interrupt Option.

Circuitry:
Standard one megacycle 3C S-PAC digital modules

Signal Levels:
Zero volts for logical ZERO; -6 volts for logical ONE. All inputs are diode/coupled/isolated; all output circuits are clamped.

Dimensions:
7'8" x 3'3" x 5'1" high

Weight:
2,000 lbs.

Power:
1,400 watts, single phase 115 + 10 volt, 58 to 62 cps; power failure protection.

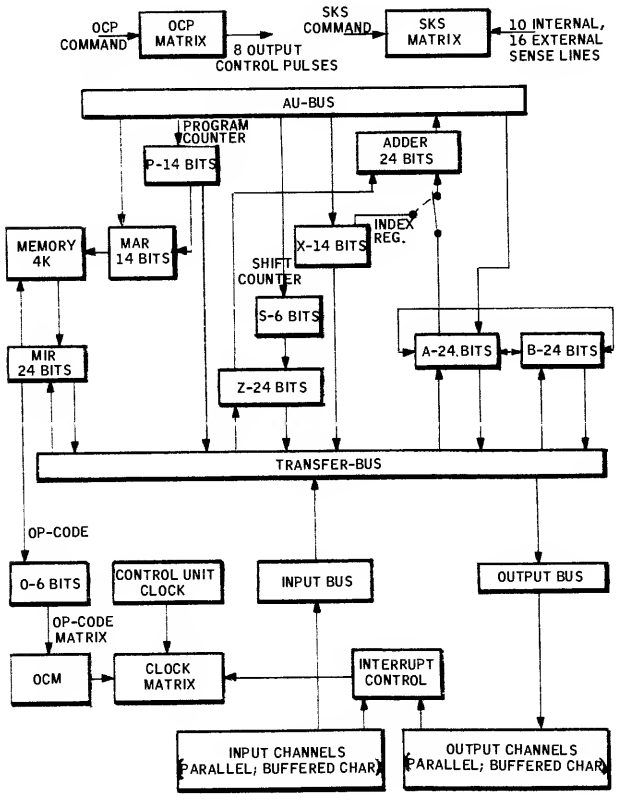


Figure 2 — DDP-24 Orangization

COMPUTER ORGANIZATION

Functionally, the DDP-24 consists of four units:

1) Arithmetic Unit
2) Control Unit
3) Input-Output Unit
4) Memory Unit

The arithmetic unit contains the arithmetic registers and the adder, in which the manipulation of data takes place during the execution of the computer program. The control unit controls the execution of the computer program by providing the timing of successive fetches from memory commands and operands as called for by the program; it also controls the execution of the computer commands. The input-output section controls the data transfers between the processor and the external system. The memory unit consists of a core memory with a 24-bit information buffer register (MIR) and a 14-bit memory address register (MAR).

ARITHMETIC UNIT

Contains the A-register, B-register, and Adder. The Z-register may also be considered part of the arithmetic unit at certain times.

A-Register. The main arithmetic 24-bit register.

B-Register. The auxiliary arithmetic 24-bit register.

Z-Register. A 24-bit register receiving its information from the memory.

After a command fetch the 14 least significant bits contain the address of the instruction, or the 14 bit operand in some cases. After an operand fetch the Z-register contains the operand and then is to be considered part of the arithmetic unit.

Adder. The logic structure providing for sums or differences during computations.

CONTROL UNIT

Consists of the program counter (P-register), op-code register (O-register), op-code matrix (OCM), index register (X-register), shift counter (S-register), transfer bus, AU bus, control unit clock and clock matrix. In addition, the Z-register can sometimes be considered part of the control unit, sometimes part of the arithmetic unit. The functions of these registers are:

Program Counter (P). Contains the memory location of the next instruction to be performed. Normally its contents are incremented by a ONE each time a new command is accessed from memory. In case of a transfer, the program counter will be loaded with the memory location to which the program shall transfer. P contains 14 bits.

Op-code Register (O). Contains the 6 bit op-code for the instruction being performed, the index bit and the indirect address bit.

Op-code Matrix (OCM). This is a gating structure, decoding the 6-bit op-code into different instruction controls.

Index Register (X). If the 14-bit index register is specified by an instruction, its contents are added to the address of the instruction and the result becomes the effective address. There are a few instructions, such as scale and normalize, where the index register is used for control and counting.

Shift Counter (S). This is used with the shift instructions, indicating the number of shift steps. S contains 6 bits.

Transfer Bus and AU Bus. These are two gating structures to which computer registers are connected both as source and destination. In this way transfer of information from one register to another can take place by the simultaneous gating of the proper paths from one register to the transfer bus and from the transfer bus to the other register. The transfer bus and AU bus allow simple interconnections between the many different computer registers and the adder. They provide easy and powerful monitoring of computer operations during trouble shooting and are displayed on the maintenance panel.

Control Unit Clock. This is the generator of the timing pulses needed for the different operation sequences of the command executions.

Clock Matrix. In the clock matrix the control unit clock signals are properly gated with the outputs from the op-code matrix and supply the correct sequence of pulses and signals for each command to be executed.

INPUT-OUTPUT

This section consists of:

a) The input-output channels with their channel ready and channel enable signals

b) The automatic interrupt which forces the computer program to be interrupted and jump to an interrupt destination in memory

c) The decoding of the incoming sense signals for test by the SKS instruction (Skip if Sense Line not Set)

d) the outgoing control signals of the OCP command (Output Control Pulse).

Data transfer of input-output information with the DDP computer takes place on a separate input-output transfer bus connected to the computer transfer bus. This allows maximum flexibility and expandability of DDP input-output capabilities. Input-output data transfer may take place to or from memory locations or the A-register.

MEMORY

The memory is a magnetic core unit of 4,096 24-bit words. It contains its own 24-bit information register (MIR) and address register (MAR). The memory transfers data to and from the DDP computer via the

transfer bus. The memory of the DDP-24 can optionally be expanded to 32,768 words by adding identical 4,096 memory modules.

Each module has a MIR and MAR which can operate independently of the processor under control of an optional fully buffered I/O control unit.

WORD FORMATS

The word length of the DDP-24 is 24 bits. A computer word may be either a command word or a data word; the difference is only in interpretation by the computer. A command word specifies the command to be executed and consists of an op-code which is a numeric code for the operation, and an address which generally specifies the memory cell from which the operand is to be read. For a number of commands the address portion of the command word represents the operand itself; no further reading from memory for the operand is required.

In addition to the op-code and address fields there are address modification bits: an index bit, which specifies the index register will be used with the operation, and an indirect addressing bit indicating that the indirect address mode will be used (refer to ADDRESSING MODES).

A data word is represented by a binary number. A sign bit indicates if it is positive (0) or negative (1); the remaining bits represent the numeric value. The DDP-24 uses sign-magnitude code, which means that negative numbers differ only in sign from positive numbers. The magnitude portion represents the numeric value irrespective of sign. (Further details in Appendix A.)

Also, in some cases the address portion carries information specifying additional operations, along with the op-code bits. Shift commands are examples of this feature.

Word formats are shown in Figure 3. In the command word the op-code is given by bits 4 through 9, the address by bits 11 through 24; bit 1 is for indirect address specification, bit 3 is the index bit, bits 2 and 10 are reserved for the addition of optional features.

Data words for the DDP-24 have bit 1 as a sign bit (0 is positive; 1 is negative) and the remaining bits as magnitude bits.

ADDRESSING MODES

For commands which involve an operand, an effective address in memory is to be specified for that operand. The following addressing modes are available:

1) Direct Address. The address portion only of the command word specifies the operand address. Up to 16,384 memory locations can be directly addressed. A special option will permit addressing directly to 32,768 locations.

2) Indexed Address. The effective address is the sum of the address portion of the command word and the contents of the index register, specified by the index bit.

3) Indirect Address. The effective address of the command word specifies a word stored in memory. The address portion of that word is the address of the operand. The indirect address mode is specified with an indirect address tag of ONE in the command word. Indirect addressing may be done progessively, e.g., if the address word stored in memory contains an indirect address bit of ONE itself, the operand address is to be found in the memory word specified by it, and so on.

Indexing takes precedence over indirect addressing. When both index bit and indirect address bit are specified, the effective address, containing the operand address, is the sum of the address portion of the command word and the contents of the index register. In progressive indirect addressing the effective address of each stage is determined by the indirect and index tags of the memory location from which it was fetched.

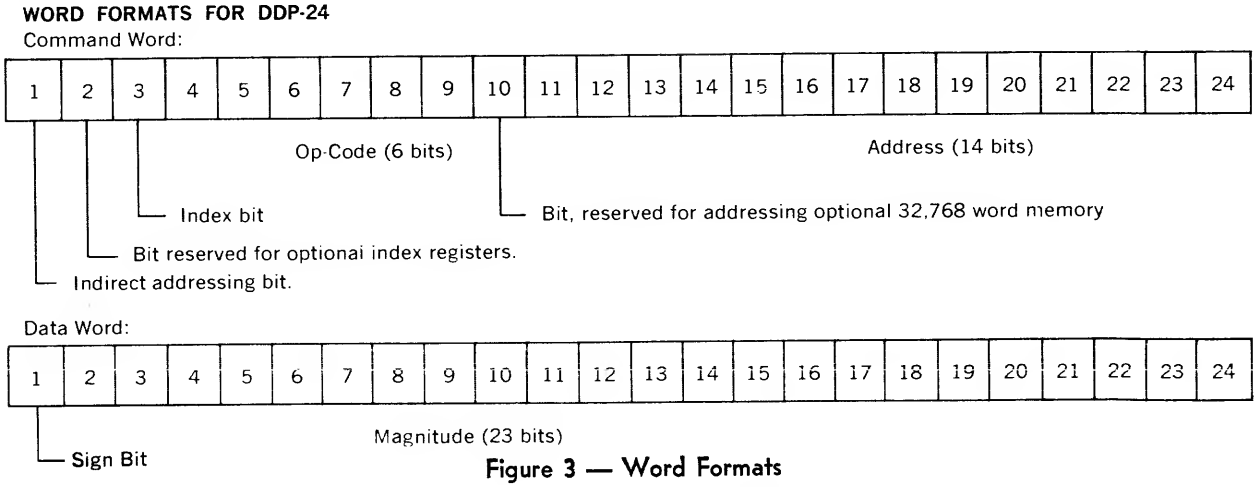


Figure 3 — Word Formats

COMMAND REPERTOIRE

LOAD AND STORE INSTRUCTIONS

Code	Mnemonic	Description	Execution Time
03	STB	Store B	10 μsec

The contents of B replace the contents of the memory word at the effective address. The contents of B are unchanged.

Code	Mnemonic	Description	Execution Time
04	STC	Store Op-Code Portion of A	10 μsec

The contents of A, bits 1-9, replace the contents of the memory word, bits 1-9, at the effective address. The contents of A and the address portion of the memory word, bits 10-24, are unchanged.

Code	Mnemonic	Description	Execution Time
05	STA	Store A	10 μsec

The contents of A replace the contents of the memory word at the effective address. The contents of A are unchanged.

Code	Mnemonic	Description	Execution Time
06	STD	Store Address Portion of A	10 μsec

The contents of A, bits 10-24, replace the contents of the memory word, bits 10-24, at the effective address. The contents of A and the op-code portion of the memory word, bits 1-9, are unchanged.

Code	Mnemonic	Description	Execution Time
23	LDB	Load B	10 μsec

The contents of the memory word at the effective address replace the contents of B. The contents of the memory word are unchanged.

Code	Mnemonic	Description	Execution Time
24	LDA	Load A	10 μsec

The contents of the memory word at the effective address replace the contents of A. The contents of the memory word are unchanged.

Code	Mnemonic	Description	Execution Time
55	TAB	Transfer A to B	5 μsec

The contents of A replace the contents of B. The contents of A are unchanged. The address portion and index bit of this instruction, bits 11-24 and 3, are not interpreted.

Code	Mnemonic	Description	Execution Time
57	IAB	Interchange A and B	10 μsec

The contents of A and B are interchanged. The address portion and index bit of this instruction, bits 11-24 and 3, are not interpreted.

Code	Mnemonic	Description	Execution Time
60	CRA	Clear A	5 μsec

The contents of A, bits 1-24, are set to zero. The address portion and index bit of this instruction, bits 11-24 and 3, are not interpreted.

ARITHMETIC INSTRUCTIONS

Code	Mnemonic	Description	Execution Time
10	ADD	Add	10 μsec

The contents of the memory word at the effective address are algebraically added to the contents of A, and the resultant sum replaces the contents of A.

Code Mnemonic Description Execution Time

Overflow is possible and will set the overflow indicator. If the magnitude of the result is zero, the initial sign of A is unchanged. The contents of B and the memory word are unchanged.

11 SUB Subtract 10 μsec

The contents of the memory word at the effective address are algebraically subtracted from the contents of A, and the resultant difference replaces the contents of A. Overflow is possible and will set the overflow indicator. If the magnitude of the result is zero, the initial sign of A is unchanged. The contents of B and the memory word are unchanged.

20 ADM Add Magnitude 10 μsec

The magnitude of the contents of the memory word at the effective address are added to the contents of A, and the resultant sum replaces the contents of A. The sign of the memory word is ignored; if the sign of A is negative, a subtractive process will occur. Overflow is possible and will set the overflow indicator. If the magnitude of the result is zero, the initial sign of A is unchanged. The contents of B and the memory word are unchanged.

21 SBM Subtract Magnitude 10 μsec

The magnitude of the contents of the memory word at the effective address are subtracted from the contents of A, and the resultant difference replaces the contents of A. The sign of the memory word is ignored; if the sign of A is negative, an add will occur. Overflow is possible and will set the overflow indicator. If the magnitude of the result is zero, the initial sign of A is unchanged. The contents of B and the memory word are unchanged.

30 SMP Step Multiple Precision 10 μsec

The contents of the memory word at the effective address are added to or subtracted from A such that the result has the sign of the result of the overall multiple precision operation. This sign and the selection of either add or subtract is determined by the instruction executed prior to the SMP instruction. Normally, this will take place at the beginning of a multiple precision routine. The add or subtract operation is for initial set-up of

Code Mnemonic Description Execution Time

of the multiple precision routine only; the sum or difference is not to be used further. For a multiple precision add, an ADD operation of the highest order portion of the two operands will be followed by SMP instructions which add all portions, starting with the lowest order and producing the same signs. For a multiple precision subtract, a SUB operation of the highest order portion of the two operands will be followed by SMP instructions which subtract all portions, starting with the lowest order and producing the same signs.

Any carry (or borrow) produced by an SMP step will be properly added (or subtracted) at the following SMP. Overflow is set by the SMP command if a carry is produced; overflow is reset if no carry is produced. Overflow of a multiple precision addition or subtraction can be detected by checking the overflow indicator after completion of the operation (normally it would not be set after the last SMP operation).

34 MPY Multiply 31 μsec

The contents of B are multiplied by the contents of the memory word at the effective address. The 23 most significant bits of the 46-bit product replace the contents of A, bits 2-24; the least significant bits replace the contents of B, bits 2-24. The signs of A and B are set to the algebraic sign of the product. The contents of A are cleared at the start of this instruction. The contents of the memory word are unchanged. The B-register must be loaded prior to the execution of the MPY instruction.

35 DIV Divide 33 μsec

The contents of the memory word at the effective address (the divisor) are divided into the contents of both A and B (the double-length dividend). The 23-bit quotient replaces the contents of B, bits 2-24; the 23-bit remainder (absolute value) replaces the contents of A, bits 2-24. The signs of A and B are set to the algebraic sign of the quotient. If the initial magnitude of A is equal to or greater than the magnitude of the memory word, the improper divide indicator is set. The contents of the memory word are unchanged.

Code	Mnemonic	Description	Execution Time
36	BCD*	BCD to Binary Conversion	33 μsec

The contents of the memory word at the effective address are converted from BCD into binary, the result replaces the contents of A. The contents of B are destroyed; the contents of the memory word are unchanged. The maximum BCD number which can be converted with this instruction is decimal +799,999.

Code	Mnemonic	Description	Execution Time
37	BIN*	Binary to BCD Conversion	33 μsec

The contents of the memory word at the effective address are converted from binary to BCD code; the result replaces the contents of B. The conversion will be performed only on those bits of the memory word which will produce a BCD code within the capacity of the B-register (24 bits). The improper divide indicator will be set if the binary number to be converted is greater than octal 3,032, 377, resulting in a BCD number greater than decimal 799,999. The contents of A are destroyed; the contents of the memory word are unchanged.

Code	Mnemonic	Description	Execution Time
62	RND	Round A	6 μsec

The contents of A are incremented by one if bit 2 in the B register is a one; the contents of A are unchanged if bit 2 (in B) is a zero. The address portion and index bit of this instruction, bits 11-24 and 3, are not interpreted. Overflow is possible and will set the overflow indicator. The contents of B remain unchanged.

LOGICAL INSTRUCTIONS

Code	Mnemonic	Description	Execution Time
15	ANA	AND to A	10 μsec

This instruction forms the logical product of the contents of A bits 1-24, and the contents of the memory word at the effective address and replaces the contents of A with the result. For each ZERO in the contents of the memory word, a ZERO is written into the corresponding bit in A; for each ONE in the memory word, the corresponding bit in A is unchanged. The contents of B and the memory word are unchanged.

* Optional Command

Code	Mnemonic	Description	Execution Time
16	ORA	OR to A	10 μsec

This instruction forms the logical sum of the contents of A, bits 1-24, and the contents of the memory word at the effective address and replaces the contents of A with the result. For each ONE in the contents of the memory word, a ONE is written into the corresponding bit in A; for each ZERO in the memory word, the corresponding bit in A is unchanged. The contents of B and the memory word are unchanged.

Code	Mnemonic	Description	Execution Time
17	ERA	Exclusive OR to A	10 μsec

This instruction forms the logical exclusive sum of the contents of A, bits 1-24, and the contents of the memory word at the effective address and replaces the contents of A with the result. For each ONE in the contents of the memory word, the corresponding bit in A is complemented; for each ZERO in the memory word, the corresponding bit in A is unchanged. The contents of B and the memory word are unchanged.

SHIFT INSTRUCTIONS

Code	Mnemonic	Description	Execution Time
40	ARS	A Right Shift	5 + n μsec

The contents of A, bits 2-24, are shifted to the right the number of positions specified by the six least significant bits of the instruction, bits 19-24. The sign of A is not shifted and is unchanged. ZEROs are shifted into the vacated position next to the sign of A, bit 2; bits shifted out of the low order position are lost. This instruction may be indexed, in which case the number of shift steps is the sum of the address portion of the instruction and the contents of the index register. The contents of B are unchanged.

Code	Mnemonic	Description	Execution Time
41	ALS	A Left Shift	5 + n μsec

The contents of A, bits 2-24, are shifted to the left the number of positions specified by the six least significant bits of the instruction, bits 19-24. The sign of A is not shifted and is unchanged. ZEROs are shifted into the vacated low order position of A; bits shifted out of the position next to the sign of A (bit 2) are lost. This instruction may be indexed, in which case the number of shift steps is the sum of the address portion of the

Code	Mnemonic	Description	Execution Time

instruction and the contents of the index register. The contents of B are unchanged.

42 LRR Long Right Rotate 5 + n μsec

The contents of A, bits 1-24 and B, bits 1-24, are treated as a single 48-bit register and are rotated to the right (end around carry) the number of positions specified by the six least significant bits of the instruction, bits 19-24. The signs of A and B are also shifted. Bits shifted out of the low order position of A enter the high order position of B; bits shifted out of the low order position of B enter the high order position of A. This instruction may be indexed, in which case the number of shift steps is the sum of the address portion of the instruction and the contents of the index register.

43 LLR Long Left Rotate 5 + n μsec

The contents of A, bits 1-24, and B, bits 1-24, are treated as a single 48-bit register and are rotated to the left (end around carry) the number of positions specified by the six least significant bits of the instruction, bits 19-24. The signs of A and B are also shifted. Bits shifted out of the high order position of B enter the low order position of A; bits shifted out of the high order positon of A enter the low order position of B. The instruction may be indexed, in which case the number of shift steps is the sum of the address portion of the instruction and the contents of the index register.

44 LRS Long Right Shift 5 + n μsec

The contents of A, bits 2-24, and B, bits 2-24, are treated as a single 46-bit register and are shifted to the right the number of positions specified by the six least significant bits of the instruction, bits 19-24. The signs of A and B are not shifted; however, the sign of B is made to agree with the sign of A. ZEROs are shifted into the vacated position next to the sign of A, bit 2; bits shifted out of the low order position of A enter the position next to the sign of B, bit 2. Bits shifted out of the low order position of B are lost. This instruction may be indexed, in which case the number of shift steps is the sum of the address portion of the instruction and the contents of the index register.

Code	Mnemonic	Description	Execution Time

45 LLS Long Left Shift 5 + n μsec

The contents of A, bits 2-24, and B, bits 2-24, are treated as a single 46-bit register and are shifted to the left the number of positions specified by the six least significant bits of the instruction, bits 19-24. The signs of A and B are not shifted; however, the sign of A is made to agree with the sign of B. ZEROs are shifted into the vacated low order position of B; bits shifted out of the position next to the sign of B, bit 2, enter the low order position of A. Bits shifted out of the position next to the sign of A, bit 2, are lost. This instruction may be indexed, in which case the number of shift steps is the sum of the address portion of the instruction and the contents of the index register.

46 NRM Normalize 5 + n μsec

The contents of A, bits 2-24, and B, bits 2-24, are treated as a single 46-bit register and are shifted left until a ONE is shifted into the position next to the sign of A, bit 2, or until the contents of B replace the contents of A (46 steps). If the index position, bit 3, is a ONE, the number of shifts required for normalization is subtracted from the index register. If the index position, bit 3, is a ZERO, the index register is not affected by this instruction. ZEROs are shifted into the vacated low order position of B; bits shifted out of the position next to the sign of B, bit 2, enter the low order position of A.

If a ONE is in the position next to the sign of A, bit 2, at the start of the operation (already normalized), the instruction will be treated as a NOP. The signs of A and B are not shifted and are unchanged.

47 LGL Logical Left Shift 5 + n μsec

The contents of A, bits 1-24, are shifted to the left the number of positions specified by the six least significant bits of the instruction, bits 19-24. The sign of A is also shifted. ZEROs are shifted into the vacated low order position of A; bits shifted out of the high order position of A are lost. This instruction may be indexed, in which case the number of shift steps is the sum of the address portion

Code	Mnemonic	Description	Execution Time

of the instruction and the contents of the index register. The contents of B are unchanged.

64 SCR Scale Right 5 + n μsec

The contents of A, bits 2-24, and B, bits 2-24, are treated as a single 46-bit register and are shifted to the right the number of positions specified by the six least significant bits of the instruction, bits 19-24. The number of positions shifted will be added to the contents of the index register. The signs of A and B are not shifted; however the sign of B is made to agree with the sign of A. ZEROs are shifted into the vacated position next to the sign of A, bit 2; bits shifted out of the low order position of A enter the position next to the sign of B, bit 2. Bits shifted out of the low order position of B are lost. This instruction is not valid if the index position, bit 3, is a ZERO.

65 SCL Scale Left 5 + n μsec

The contents of A, bits 2-24 and B, bits 2-24, are treated as a single 46-bit register and are shifted to the left the number of positions specified by the six least significant bits of the instruction, bits 19-24. The number of positions shifted will be subtracted from the contents of the index register. The signs of A and B are not shifted; however, the sign of A is made to agree with the sign of B. ZEROs are shifted into the vacated low order position of B; bits shifted out of the position next to the sign of B, bit 2, enter the low order position of A. Bits shifted out of the position next to the sign of A, bit 2, are lost. This instruction is not valid if the index position, bit 3, is a ZERO.

JUMP INSTRUCTIONS

12 SKG Skip if A Greater 10 or 12 μsec

The contents of A are algebraically compared to the contents of the memory word at the effective address. If the value in A is greater than the value in the memory word, the next instruction is skipped and the computer resumes at that point. If the value in A is equal to or less than the value in the memory word, the computer takes the next sequential instruction. The contents of A and the memory word are unchanged.

13 SKN Skip if A Not Equal 10 or 12 μsec

The contents of A are algebraically compared to the contents of the memory word at the effective address. If the value in A is equal to or greater than the value in the memory word, the next instruction is skipped and the computer resumes at that point. If the value in A is less than the value in the memory word, the computer takes the next sequential instruction. The contents of A and the memory word are unchanged.

25 JRT Jump Return 10 μsec

This instruction is an indirect jump. A jump is executed to the location specified by the address portion of the memory word (bits 11-24) at the effective address. The contents of the memory word and the contents of A and B are unchanged. This instruction must be used for returning from all interrupt subroutines to restore the interrupt capability again (in the standard DDP-24), or to effect return to the previous priority level if optional Priority Interrupt is used.

27 JST Jump and Store Location 10 μsec

The location of the JST instruction plus one replaces the contents of the address portion of the memory word, bits 11-24, at the effective address. A jump is then executed to one location beyond the effective address. Bits 1-10 of the contents of the memory word are unchanged; the contents of A and B are unchanged. This instruction may be used for entering a subroutine.

70 JPL Jump if A Zero 6 μsec

If the sign of A, bit 1, is positive (ZERO) or if the magnitude positions of A, bits 2-24, are all ZEROs, the computer takes its next instruction from the memory word at the effective address and continues from there. If the sign of A, bit 1, is negative (ONE) and one or more of the magnitude positions of A, bits 2-24, are ONEs, the computer takes the next sequential instruction. Thus, a jump for A negative could be accomplished by an unconditional jump instruction (JMP)

Code	Mnemonic	Description	Execution Time

immediately following the JPL instruction. The contents of A are unchanged.

71 JZE Jump if A Zero 5 μsec

If all of the magnitude positions in A, bits 2-24 are ZEROs, the computer takes its next instruction from the memory word at the effective address and continues from there. If any of the magnitude positions of A are ONEs, the computer takes the next sequential instruction. The sign of A, bit 1, is ignored. The contents of A are unchanged.

73 JOF Jump on Overflow 5 μsec

If the overflow indicator is set, it will be reset and the computer will take its next instruction from the memory word at the effective address and continue from there. If the overflow indicator is not set, the computer takes the next sequential instruction. To reset the overflow indicator without altering the normal sequence of instructions, the JOF instruction may be used with an effective address that is one location greater than the address of the JOF instruction.

74 JMP Unconditional Jump 5 μsec

The computer takes its next instruction from the memory word at the effective address and continues from there. The JMP instruction with indirect addressing may be used for returning from subroutines which are not interrupt routines.

INDEX INSTRUCTIONS

54 ADX Add to Index 5 μsec

The contents of the address portion of this instruction, bits 11-24, are added to contents of the index register, and the resultant sum replaces the contents of the index register. Overflow of the index register is possible, but will be ignored. If the indirect address position of the instruction, bit 1, is a ONE, the contents of the address portion of the memory word, bits 11-24, at the effective address are added to the index register. The contents of A and the memory word are unchanged. This instruction is not valid if the index position of the instruction, bit 3, is a ZERO.

56 LDX Load Index 5 μsec

The contents of the address portion of this instruction, bits 11-24, replace the contents of the index register. If the indirect address position of the instruction, bit 1, is a ONE, the contents of the address portion of the memory word, bits 11-24, at the effective address replace the contents of the index register. The contents of A or the memory word are unchanged. This instruction is not valid if the index position of the instruction, bit 3, is a ZERO.

63 TAX Transfer A to Index 5 μsec

The address portion of A, bits 11-24, replace the contents of the index register. The contents of A are unchanged. This instruction is not valid if the index position of the instruction, bit 3, is a ZERO. The indirect address position and the address portion of this instruction, bits 1 and 11-24, are not interpreted.

66 STX Store Index 10 μsec

The contents of the index register replace the contents of the address portion of the memory word, bits 11-24, at the effective address. The contents of A and the index are unchanged; bits 1-10 of the memory word are unchanged. This instruction is not valid if the index position of the instruction, bit 3, is a ZERO.

67 IRX Increment, Replace and Load Index 14 μsec

The contents of the address portion of the memory word, bits 11-24, at the effective address are incremented by one. If the index position of this instruction, bit 3, is a ONE, the resulting sum replaces the contents of the address portion of the memory word and the index register. The contents of A are unchanged. If the index position of this instruction, bit 3, is a ZERO, the address portion of the memory word will be incremented and this incremented value will replace the contents of A. In this case, the index register contents are unchanged. Any carry from bit 11 is ignored. Bits 1-10 of the memory word are unchanged. Thus, it is possible to have many "index registers" in memory that can be incremented, saved and made available for use in indexing operations all with one instruction.

72 JIX Jump on Index 5 μsec

If the contents of the index register are not ZERO, the computer takes its next

Code Mnemonic Description Execution Time

instruction from the memory word at the effective address and continues from there. If the contents of the index register are ZERO, the computer takes the next sequential instruction. This instruction is not valid if the index position of the instruction, bit 3, is a ZERO.

75 JXI Jump on Index Incremented 7 μsec

The contents of the index register are incremented by one and the resulting sum replaces the contents of the index register. If this resulting sum is not ZERO, the computer takes its next instruction from the memory word at the effective address and continues from there. If the sum is ZERO, the computer takes the next sequential instruction. This instruction is not valid if the index position of the instruction, bit 3, is a ZERO.

INPUT-OUTPUT INSTRUCTIONS

07 INM Input to Memory 10 μsec

The input word from a previously enabled input channel (refer to OCP) replaces the contents of the memory word at the effective address.

22 OTM Output from Memory 10 μsec

The contents of the memory word at the effective address are transferred as output to the previously enabled output channel (refer to OCP). The contents of the memory word are unchanged.

31 FMB Fill Memory Block variable

This instruction is used for high speed input into the block of consecutive memory locations starting with the memory word at the effective address. Once started, the sequence continues without interruption, controlled asynchronously by an external ready signal. The FMB instruction may operate with any input channel that has been previously enabled (refer to OCP). The contents of the index register are incremented by one for each word being stored thereby increasing the effective address. Execution of this instruction may be terminated by either an external signal (e.g., a stop code) or upon the contents of the index register having become all ZEROs. This instruction is not valid if the index position, bit 3, is a ZERO. The FMB instruction can process input data at a 166 kc word rate.

Code Mnemonic Description Execution Time

32 DMB Dump Memory Block variable

This instruction is used for high speed output from the block of consecutive memory locations starting with the memory word at the effective address. Once started, the sequence continues without interruption, controlled asynchronously by an external ready signal. The DMB instruction may operate with any output channel that has been previously enabled (refer to OCP). The contents of the index register are incremented by one for each output word being transferred thereby increasing the effective address. Execution of this instruction may be terminated by either an external signal or upon the contents of the index register having become all ZEROs. This instruction is not valid if the index position, bit 3, is a ZERO. The DMB instruction can process output data at a 166 kc word rate.

50 OTA Output from A 5 μsec

The contents of the A register are transferred as output to the previously enabled output channel. If bit 11 of this instruction contains a ONE, bits 19-24 form a 6 bit mask. This provides a facility for flexibly formatting of character outputs. For ZERO mask bits, there will be corresponding ZERO bits in the output; for ONE mask bits, the corresponding bits in A will be the output. If bit 11 contains a ZERO, 24-bit output words are transferred by this instruction. The index position of the instruction, bit 3, is not interpreted. The contents of A are unchanged.

51 ITC Interrupt Control 5 μsec

Interrupt is enabled if bit 11 of this instruction contains a ONE; interrupt is inhibited if bit 11 contains a ZERO. The index position and the remaining positions of the address portion of the instruction, bits 3 and 12-24 are not interpreted.

52 INA Input to A 5 μsec

The input word from a previously enabled input channel (see OCP) replaces the contents of A. If bit 11 of this instruction contains a ONE, bits 19-24

Code Mnemonic Description Execution Time

form a 6 bit mask. This provides a facility for flexibly formatting of character inputs. For ZERO mask bits, there will be corresponding ZERO bits in A; for ONE mask bits, the corresponding input bits will replace the contents of A. If bit 11 contains a ZERO, 24-bit input words are transferred by this command. The index position of the instruction, bit 3, is not interpreted.

53 OCP Output Control Pulse 5 μsec

An output pulse is generated by this instruction for the control of input-output channels and external equipment. The address portion, bits 11-24, specifies the unit to be selected, the type of control, etc. (refer to APPENDIX C for the code assignments). The index position of the instruction, bit 3, is not interpreted.

61 SKS Skip if Sense Line Not Set 5 μsec

The sense line specified by the address portion of this instruction, bits 11-24, is interrogated. If the sense line is not set, the computer skips the next instruction and continues from there; if the sense line is set, the computer will take the next sequential instruction. The lines that may be tested include 10 internal sense lines (six sense switches, overflow indicator, improper divide indicator, input parity and stop code), ready signals of input-output channels and external sense lines from peripheral equipment (busy status, parity errors, etc.). From one to ten of the internal sense lines may be tested simultaneously; in which case any or all of the tested sense lines may cause a skip. For the channel-ready sense lines, similar simultaneous testing is also possible. If the index position of the instruction, bit 3, is a ONE, the flip flop associated with the tested sense line is reset. APPENDIX D contains the sense line selection assignments.

Code Mnemonic Description Execution Time

CONTROL INSTRUCTIONS

00 HLT Halt 5 μsec

The computer will halt until the START button is manually depressed (see description of operation), at which time execution will be resumed at the next sequential instruction. The indirect address position, index position and address portion of this instruction, bits 1, 3 and 11-24, are not interpreted.

02 XEC Execute 5 μsec + variable

The instruction in the memory word at the effective address is executed. After execution of the specified instruction, the computer takes the next sequential instruction following the XEC instruction and continues from there. If the executed instruction involves a jump, the computer takes its next instruction from the jump destination and continues from there; if the executed instruction involves a skip, the skip will be relative to the XEC instruction and not the instruction at the effective address.

77 NOP No Operation 5 μsec

No operation is performed by this instruction. The computer will take the next sequential instruction and continue from there. The index position and address portion of this instruction, bits 3 and 11-24, are not interpreted.

INPUT-OUTPUT

STANDARD PERIPHERAL EQUIPMENT

The DDP-24 includes an electric typewriter for input and output with a 15 character per second rate. Also supplied is a photoelectric paper-tape reader (300 characters per second) and a paper-tape punch (60 characters per second). The paper-tape reader and punch operated in either continuous or pulsed mode. In pulsed mode, the reader will stop after each character; the computer program gives a start signal when the next character can be read. The typewriter features keyboard lock which may be engaged or released under program control. This allows data input from the typewriter without interfering with other peripheral equipment. When the paper tape reader is started, the keyboard is automatically locked. Control panel switches select different on-line and off-line combinations of typewriter, paper tape reader and punch. These include off-line paper tape preparation from keyboard, type-out of paper-tape input and paper-tape duplication by punch-out of paper tape input. The typewriter off-line switch releases the keyboard lock. The input-output diagram (Figure 4) illustrates the interconnection of the standard peripheral equipment, input-output channels, interrupt control, OCP and sense inputs.

SENSE LINES

A number of single lines from either external or internal sources may be tested by the SKS command 61 (Skip if Sense Line). In the standard version the following test inputs are provided.

1) Six sense switches on the control panel, to allow the programmer or operator manual control of program branching.
2) Four internal computer flip-flops. These are overflow, improper divide, input parity, and stop code.

   NOTE: All or several of the 10 test inputs in groups 1 or 2 can be tested simultaneously as controlled by the address portion of the SKS command. Any or all of the tested sense lines may cause a skip.
3) Four sense lines from the ready flip-flops of the 4 standard I/O channels of which all or several can be tested simultaneously, under control of the address portion of the SKS command. Ready signals of any optional additional I/O channels may also be tested simultaneously; up to 12 channel ready flip-flops altogether. Any or all of the tested channel ready sense lines may cause a skip.
4) Sixteen external sense lines from peripheral equipment. Only one such line can be tested at a time. The number of test signals external to the computer can be expanded to over 2,000.

OUTPUT CONTROL PULSES

The OCP command 53 (Output Control Pulse) provides control pulses for up to 4,096 different lines to external equipment. The coding of the address portion

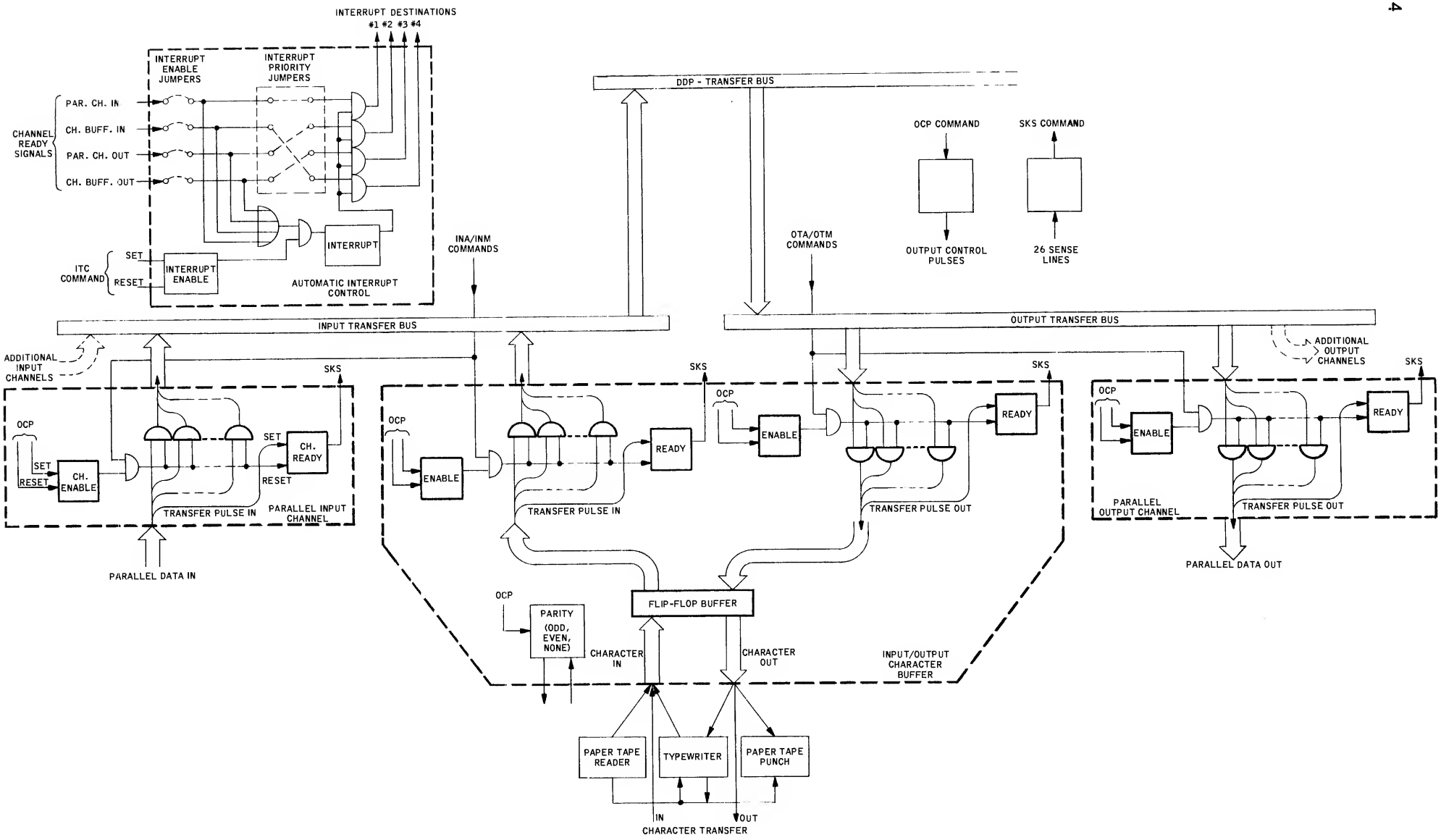


Figure 4 — I/O Block Diagram

of this command selects the line. Codes have been selected and others reserved for certain input-output control functions, such as start and stop devices and enabling input-output control channels. The standard DDP-24 provides 8 general purpose output control pulses for external use. Over 4,000 lines can be added.

Three types of control signals are available:

1) 2 μsec output pulses. These lines are available in groups of four. The OCP pulse is generated by the execution of a properly coded OCP instruction.
2) DC output lines available in groups of four. The DC levels are set and reset under control of OCP-commands.
3) DC output lines with power drivers. These operate the same as 2 above, except a power driver is included for setting relays and other heavy load applications.

PARALLEL INPUT-OUTPUT CHANNELS

One parallel input channel and one parallel output channel are provided with the DDP-24. These allow for full 24-bit parallel information transfer into or out of the computer. The two parallel channels handle inputs or pulse outputs. A ready signal is provided to indicate when the channel can transfer data; a channel-enable flip-flop enables the channel for subsequent information transfer into or out of the processor. The channel enable flip-flops are set by the computer with properly coded OCP commands. Parallel channels are used with parallel peripheral devices, such as A-D or D-A converters, printers and Digital Resolvers. Additional parallel channels are optionally available — either for input or output, with or without a flip-flop buffer. The optional input-output expansion capabilities of the DDP-24 are shown in Figure 5.

CHARACTER BUFFER

One 6-bit character buffer is provided for both input and output characters. The character buffer includes a 6 bit flip-flop register; input channel and output channel; parity detection for inputs, parity generation for outputs, stop code flip-flop, a character input-ready signal, a character output-ready signal, a character buffer input enable flip-flop, and a character buffer output enable flip-flop. Odd parity is used, except for the typewriter which does not have parity. The stop code flip-flop is set by a stop code from paper tape, or keyboard, and can be tested by the SKS command.

The typewriter, paper-tape reader, and paper-tape punch are normally connected to the input-output character buffer. Other character devices, such as magnetic tape units, can also be connected. Devices can time share the character buffer under program control. Only one device can operate with the character buffer at a time.

Additional character buffers, to allow several character devices to operate simultaneously, are optional. These may be either for character inputs, character outputs, or both character input and output. An overall input parity signal is provided when additional input character buffers are used. The signal operates the control panel parity light.

AUTOMATIC INTERRUPT

Any standard parallel channel or character buffer is capable of operating either in the automatic interrupt or ready mode under control of a jumper in the input-output section.

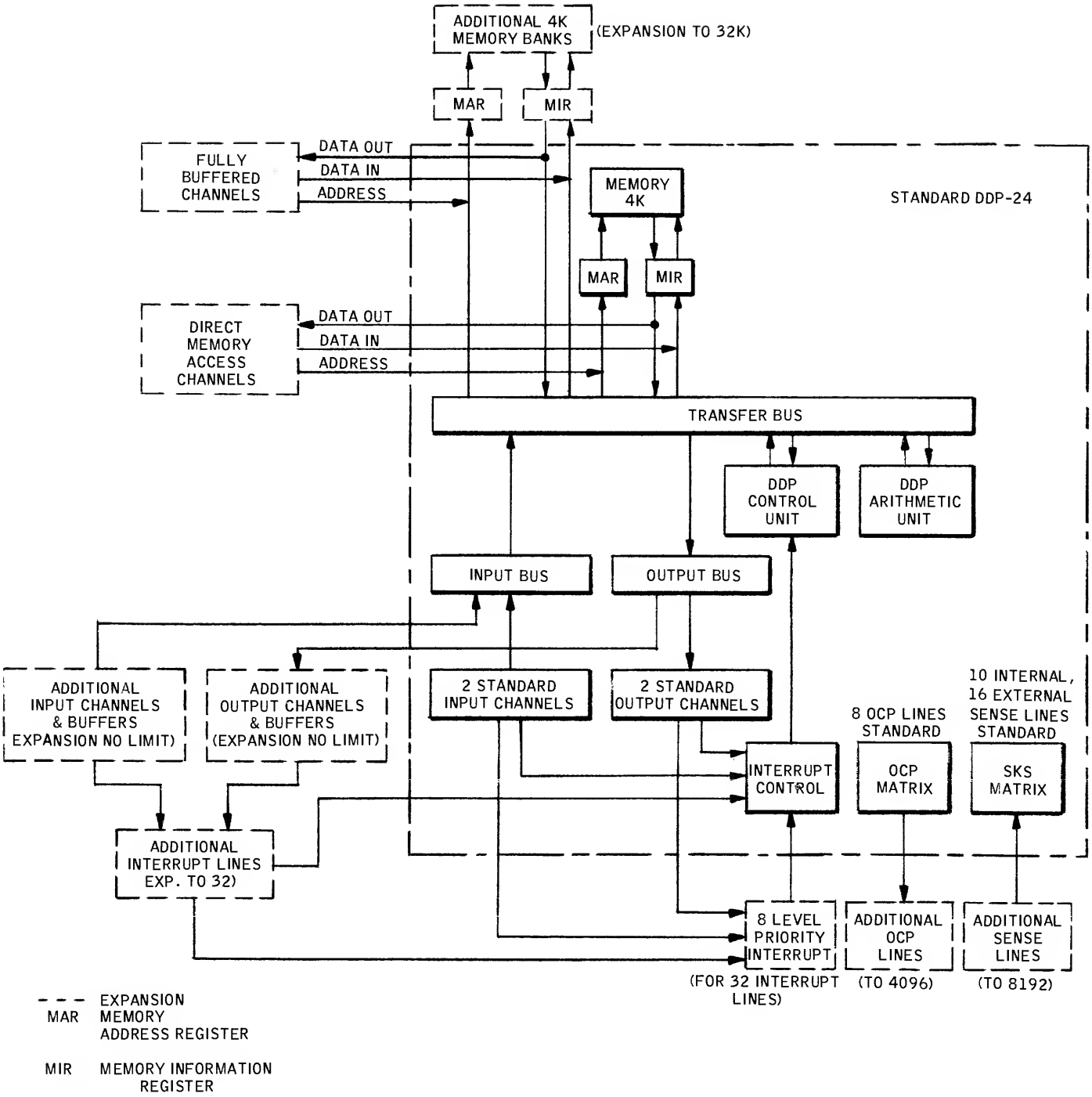
In the interrupt mode, the computer program is automatically interrupted when a channel is ready to transfer data. Branching to the proper subroutine takes place automatically while the contents of the program counter at interrupt are stored. On completion of the interrupt subroutine the main program resumes at the interrupted point. In the ready mode, the program tests for the channel-ready signal with periodic SKS commands. If the ready signal is not present, the computer program proceeds with the current routine and later repeats the test. The program may also branch into a test loop until the ready signal is available. If the test of the ready signal is successful the program will transfer to the input or output subroutine.

The overflow flip-flop and improper divide flip-flop can be connected to initiate an interrupt when set. The circuit for power failure protection will generate an interrupt if power falls below a threshold value.

The DDP-24 is equipped with four interrupt destinations. If an interrupt is initiated by either of the two parallel channels, the character input channel or the character output channel, an automatic program jump takes place to a different memory location depending on the channel. The fourth interrupt destination is shared by the character buffer out, the overflow flip-flop and the improper divide flip-flop. If several channels with different interrupt destinations cause an interrupt at the same time, a preassigned priority prevails, such that the interrupt from a channel with a high order priority is acknowledged first.

For additional I/O channels in Automatic Interrupt mode, interrupt lines can be added each with a unique interrupt destination. Up to 32 interrupt lines and destinations are standard options with the DDP-24.

In the standard processor additional interrupts will be inhibited during the execution of an interrupt subroutine. A second interrupt is handled immediately after completion of the current subroutine. The interrupt inhibit flip-flop is reset by the execution of a JRT command. Any subroutine which may be part of an



EXPANSION DIAGRAM
STANDARD DDP-24

Figure 5 — I/O Expansion Diagram

interrupt subroutine must not use a JRT command for subroutine return, but an indirectly addressed JMP instruction.

An automatic priority control system can be added. With this system interrupt of interrupt is possible. The priority system permits assignment of up to 8 levels of priority to input-output channels, and/or parts of the main program. An interrupt by an input or output channel will occur when its priority is higher than the program in progress. (See Priority Interrupt.)

If an interrupt occurs while the computer is in HALT condition, the interrupt subroutine will be executed as normal; upon completion, the computer will return to HALT again with the same value in the program counter as before.

The transfer rate for channels operating in the interrupt mode depends on the length of the input-output subroutine. For a channel in the automatic interrupt mode transfer rates over 25,000 words are possible. Interrupt lines can be provided to interrupt the computer program without transfer of data, such as with an alarm condition. The interrupt capability of the DDP can be either enabled or inhibited by the TIC command. The masterclear button on the control panel will disable interrupt to prevent runaway interrupt conditions.

INPUT-OUTPUT DATA TRANSFER

The computer will operate with many peripheral devices connected to its standard input-output channels or to optimally added channels.

In general, more than one input or output device may be connected to an input or output channel. However, not more than one device per channel should be operating simultaneously. For simultaneous operation of several independent devices a corresponding number of input or output channels should be provided; however, simultaneous output to the paper-tape punch and typewriter via the standard I/O character buffer is possible.

When an input-output device is ready for data transfer it sets the ready flip-flop of the channel to which it is connected. If in ready mode, the computer may detect this condition with a periodic SKS instruction as part of the main program. If in interrupt mode, this SKS instruction is not necessary unless more than one channel could cause an interrupt to the same interrupt destination. The input-output subroutine will enable the channel by setting the enable flip-flop with the properly coded OCP command.

The four functional groups of input-output data channels are input channels in ready mode, input channels in interrupt mode, output channels in ready mode and output channels in interrupt mode. The enabling of a channel automatically reset any other enable flip-flop in the same group of channels. Any ready mode channel with a set enable flip-flop will effectively be disabled during an interrupt subroutine; interrupt channels will effectively be disabled when not in an interrupt subroutine.

After enabling the desired channel with an OCP command the data transfer is accomplished with either INM (Input to Memory), INA (Input to A), OTM (Output from Memory), OTA (Output from A), FMB (Fill Memory Block) or DPM (Dump Memory).

These commands allow for direct input-output data transfer into or from any memory location, or into or from the A-register. The latter is important if further handling of the input-output data is required.

The DDP program may command a data transfer with INA, INM, OTA, OTM, FMB or DMB without testing for the ready signal of the enabled channel. If the ready signal is not true, the command waits until it becomes true. These input-output commands reset the ready flip-flop of the enabled channel.

The address portion of the INA and OTA commands may be used as a mask for characters of up to 6 bits, if the most significant bit of the address portion is a ONE. This provides for easy and flexible formatting. With the INA command, ZERO bits in the mask will leave the corresponding bits in A unchanged; a ONE mask bit will enter the corresponding input bits into A. With the OTA command, ZERO mask bits will produce corresponding ZERO bits in the output; for a ONE mask bit the corresponding bits in A appear as output bits.

The following is a typical subroutine for input of a block of data from an input channel in interrupt mode:

Location	Op-Code	Address	Time
	Interrupt JST	00002	7
00003	JMP	a	5
a	INM	indexed address	10
a + 1	JXI	b	7
b	JRT	00002	10
			39 μsec

Upon interrupt, the contents of the program counter (P) are automatically stored, typically in memory location 00002 (octal notation) and the following command is executed from location 00003. The indicated execution time of 7 μsec for interrupt includes the automatic storing of (P). The INM command stores the input in the specified memory location. JXI increments the contents of the index register, and if not yet zero a conditional jump will take place to the location where a JRT command is stored. This will effect the return to the main program, where it had been broken off by the interrupt. Total execution time of the interrupt subroutine is 39 μsec, corresponding to a data trans-

fer rate of over 25,000 words per second. It has been assumed that only one input channel is operating in interrupt mode, therefore, the channel enable flip-flop can be set and remain set beforehand; not as part of the subroutine. In this example the index register is used only for the input routine. If the index register contents have become ZERO, further inputs from the external device are to be prevented by proper commands. The JRT command will terminate the subroutine in the same way.

BLOCK TRANSFER

Block transfers are possible on any input or output channel with the standard commands FMB (Fill Memory Block) and DMB (Dump Memory Block). During execution these commands enter or dump information in successive memory locations, directly and synchronously with the external device, until an external stop signal is given. or a limit location reached. No other instructions will be executed while a FMB or DMB is in process. The maximum transfer rate is 166,000 words per second.

The FMB or DMB commands can be used with any enabled input or output channel, and are efficient for communicating with high speed devices such as another computer, magnetic discs, drums, and so forth.

WORD FORMING BUFFERS

Optional word forming buffers for both input and output are available. This buffer permits the automatic building of a computer word out of input characters, and forming of output characters from a computer word. The buffer is composed of a 6-bit character buffer, 24-bit flip-flop register, and a buffer control. This controls:

1) the number of characters per word (either 2, 3 or 4)
2) shifting of information in the flip-flop register for the character output or input
3) the buffer ready signal, and
4) checking or generation of character parity.

In the interrupt mode the possible character transfer rate is doubled and the I/O processing time is decreased by a factor of four. A word buffer is connected to the DDP-24 with a parallel input channel and/or parallel output channel.

DIRECT MEMORY ACCESS CHANNEL

A direct memory access channel can be added which transfers input-output data directly into or out of memory without intervention by the computer program except for initial setting up of the channels.

A DMA channel consists of a 24-bit information register, which transfers data to and from the memory information register, and an address selection register, which provides address information to the memory address register, and a limit register.

This DMA unit has priority over the processor and will obtain memory access immediately when the memory completes any current read or write function. Thus, a maximum of five microseconds delay may be encountered until the memory access of the DMA channel is initiated. When such a priority access of the memory is performed, the next instruction or operand access to memory is delayed for the memory cycle required to inject the data.

The DMA can transfer data at rates up to 166,000 24-bit words per second. For rates higher than 100 kc the computer program is effectively fully interrupted; for rates lower than 100 kc the DMA channel requires only 5 μsec access to memory for each input or output. Therefore, a 50 kc data rate slows down the computer program less than 25%.

When using the channel, the DDP-24 program loads a twenty-bit block selection number into the DMA address register. The least significant fourteen bits of this selection number specify the starting address of the block to be transferred. Three bits specify which one of eight stop addresses within the memory module shall be used (00777, 01777, 02777, 03777, 04777, 05777, 06777, or 07777 for the standard 4,096 memory, in octal code). Two bits specify the memory module of the stop address, if more than 4,096 words of memory are used. The twentieth bit selects the input or output mode of operation.

Therefore, the one address selection word specifies both the start and stop address and the mode of operation.

For connection to the external equipment the DMA channel provides 24 input lines, 24 output lines and control lines.

FULLY BUFFERED CHANNEL

Fully buffered channel can be added to a DDP-24 with more than one 4,096 word memory. While the computer operates with one memory a fully buffered channel is capable of communicating directly and simultaneously with any of the other memories. Therefore, input-output operations can take place independently of and without interference with the computer program.

If the unit and the computer program address the same memory at the same time, the fully buffered channel will have priority.

Transfer rates up to 166,000 words per second are possible. The fully buffered channels include a 24-bit

data buffer register and an address selection register. The address selection register corresponds closely to the same register of the DMA and is set up initially by the DDP-24 program. It specifies the starting address of the data block to be transferred, input or output mode of operation, and the stop address to be used. For each memory module there are 8 stop addresses: x0777, x1777 x2777, x3777, x4777, x5777, x6777, and x7777, with x specifying the memory module.

The fully buffered unit includes 24 input lines, 24 output lines and control lines for connecting external equipment.

PRIORITY INTERRUPT

The optional priority interrupt system allows 8 levels of interrupt priority to be assigned to input-output channels and parts of the computer program itself. Any 4 of the 8 levels may be used for different sections of the program. Jumpers are used to assign levels of interrupt priority to I/O channels; the assignment of priority levels to parts of the main program is accomplished by properly coded OCP commands in that program. Several I/O channels or parts of the main program may have the same priority levels.

The Priority Interrupt System allows for automatic interrupts of the computer program or routine, only if the requesting priority level of the I/O channel ready for data transfer is higher than the current priority level of the I/O subroutine or main program. If a requesting priority level is lower than or equal to the current priority level an interrupt must wait until the latter has returned to a lower level.

If an interrupt is allowed, a program transfer will be effected to a memory location unique to the interrupting I/O channel, as in the standard DDP-24. From there the proper interrupt subroutine will be executed. At the end of the interrupt subroutine execution of the JRT-command effects automatic return to the priority level which existed before the interrupt occurred.

OPERATION

CONTROL PANEL

The control panel contains displays, switches and indicators necessary to operate the equipment. See Figure 6. Computer registers are displayed; switches permit manual entry of data. The control panel also includes a power switch, sense switches for program branching, and indicators and controls for standard input-output devices.

Register Display

The following registers are displayed:

1) A-register
2) B-register
3) Z-register
4) Index register
5) Program register
6) Op-code, index bit and indirect address bit

Neon push-button lights show the binary contents of the registers and allow bit by bit entry. A red reset button is provided to clear each register to all ZEROs. The reset button for the op-code register also clears the index bit and indirect address bit.

Operating Controls

Momentary push-buttons perform the following functions:

1) SINGLE OPERATION. Same as START except HALT flip-flop is not reset; computer operations stop after execution of one command.
2) START. Resumes execution of program beginning with command stored in memory location indicated by the program register.
3) STOP. Sets HALT flip-flop, prevents execution of program after current instruction is processed.
4) MASTER CLEAR. Sets HALT flip-flop, resets all displayed registers, index bit and indirect address bit flip-flops, control unit clock, ready and enable flip-flops of all input-output channels, interrupt, enable, input parity, overflow and improper divide flip-flops. Stops paper-tape reader.
5) FILL. Enters data from paper tape or typewriter without use of stored program. Characters are built into words; words stored in successive memory locations. HALT flip-flop is not reset; no program execution can take place; operation is terminated by stop-code character.

Operational Indicators

These show the following:

1) INPUT PARITY. Detection of an error while reading character information; corresponds to state of input parity flip-flop. Flip-flop can be reset with properly coded command.
2) INTERRUPT. Execution of an interrupt subroutine; corresponds to state of interrupt flip-flop.
3) OVERFLOW. An overflow indicator for detection of arithmetic and scaling errors in the program. Overflow may occur during computer operations in the following cases:
   a) For additions or subtractions which produce sums or differences surpassing the capacity of the A-register. This is valid for both

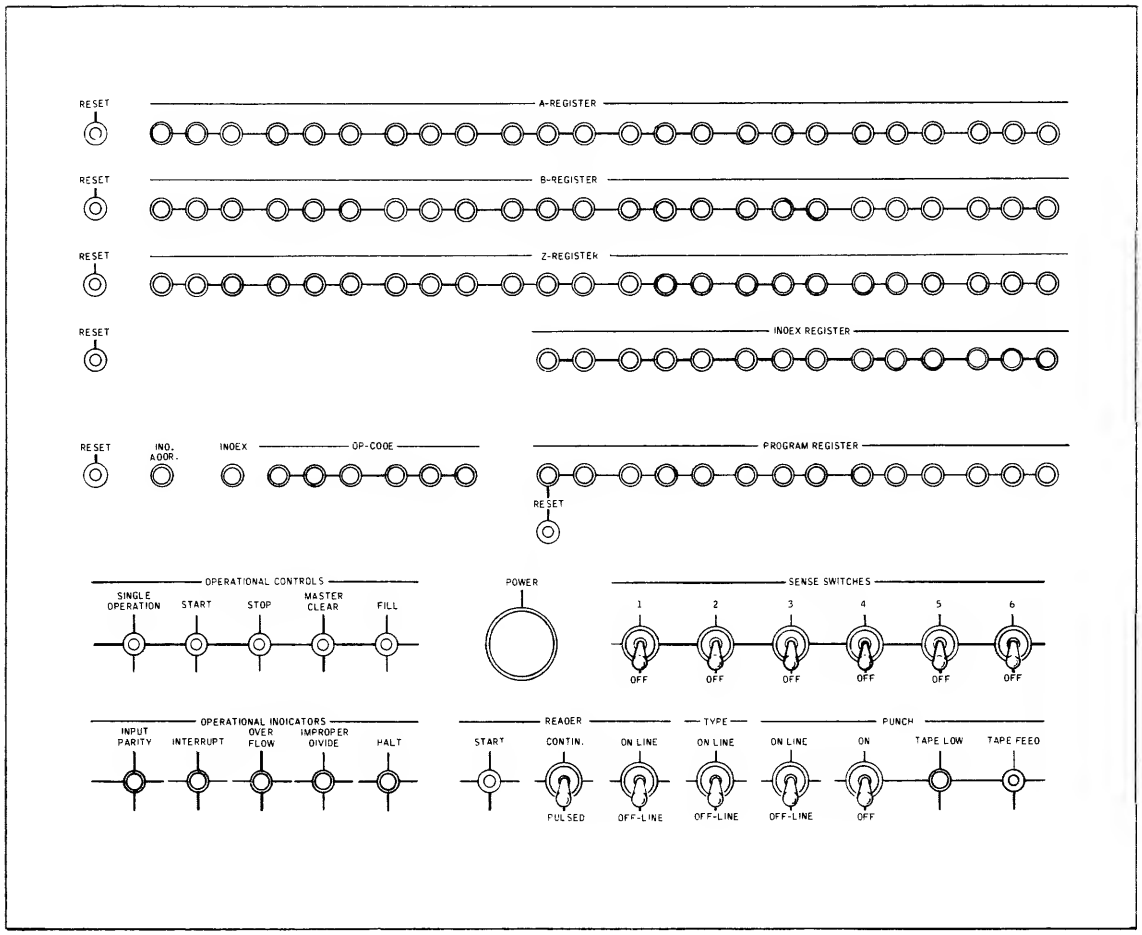


Figure 6 — Control Panel

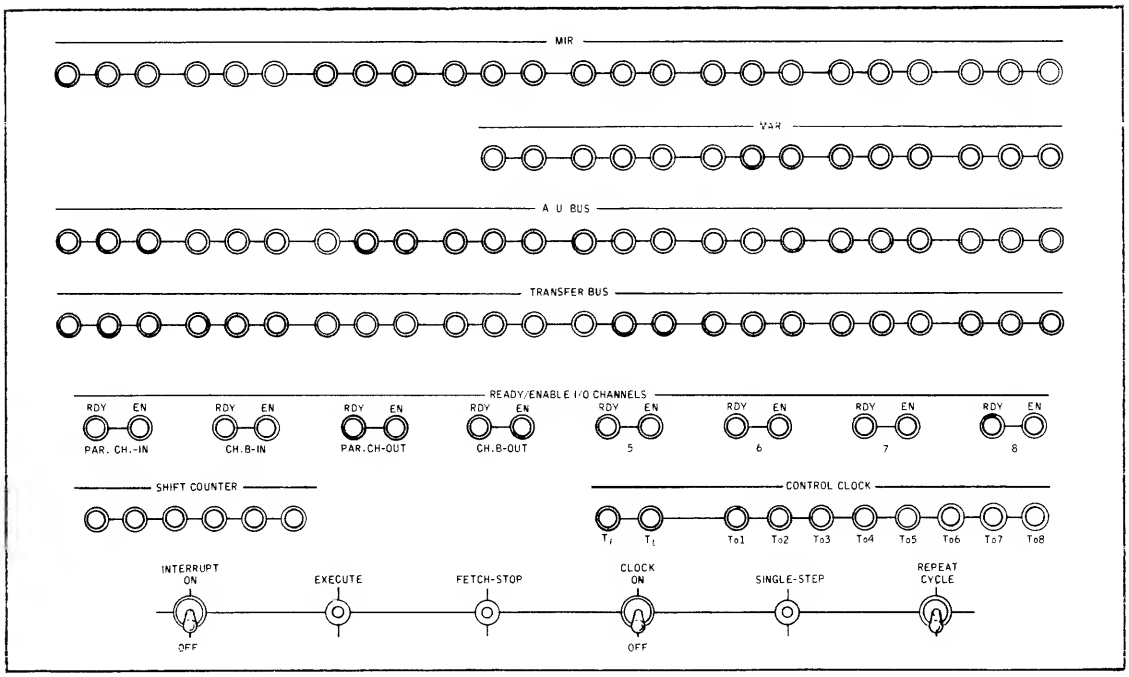


Figure 7 — Maintenance Panel

straight and absolute value additions or subtractions. Also for rounding of A with the RND-command. The overflow indicator is set by these conditions and remains set.

b) For multiple precision operations if a carry is produced by the addition or subtraction of the next lower order portion. The overflow flip-flop will be reset when a multiple precision operation step does not produce a carry, and therefore it will be reset after the complete multiple precision addition or subtraction has been performed correctly.

The overflow indicator may be reset by the Jump on Overflow command (73) and the Skip for Sense Line Not Set Command (61).

4) IMPROPER DIVIDE. The improper divide indicator is switched on in the following cases:

a) For division undertaken with a numerator larger than or equal to the denominator. The resulting quotient surpasses the capacity of the B-register.

b) For the optional Binary to BCD command, if executed with a number which upon conversion results in a BCD number greater than 799,999. The improper divide indicator may be reset by the Skip for Sense Line Not Set Command (61).

5) HALT. Halt flip-flop is set; no operations can take place except handling interrupts. On completion of the interrupt subroutine, computer returns to HALT state, with same value in program counter as before. HALT flip-flop is not reset by interrupt procedure.

Power

This self lighting pushbutton switches computer input power ON and OFF. When power is switched ON, system normalizers ensure that flip-flops are reset and the HALT condition set. This prevents undesirable runaway effects. No warm-up period is required.

Sense Switches

These provide manual branch control of the computer program. The state of any or all six switches can be tested by properly coded SKS commands, after which branching may take place.

Input-Output Controls

1) READER

a) START. Momentary pushbutton to start movement of paper tape in reader.

b) CONTIN. / PULSED. Switch allowing either continuous or pulsed tape reader operation; in pulsed mode reader stops for each character read; computer normally provides a pulse to read the next character.

c) ON LINE/OFF LINE. Provides on-or-off-

Reader	Type	Punch	On-line Program Controlled	Off-line operation
on-line	on-line	on-line	Reader, Type, Punch; normal operation	none
on-line	on-line	off-line	Reader, Typewriter	none*
on-line	off-line	on-line	Reader, Punch	Typewriter
on-line	off-line	off-line	None (reader must not be used)	Typewriter to punch, for paper tape preparation.
off-line	on-line	on-line	Type, punch	none*
off-line	on-line	off-line	None (typewriter must not be used.)	Tape reader to punch, for paper tape duplication.
off-line	off-line	on-line	None	Reader to typewriter; for printout of paper tape.
off-line	off-line	off-line	None	Reader to typewriter and punch, or type to punch

* Not of much practical significance.

Figure 8 — On-line/off-line Combinations

line reader operation; for on-line mode characters are entered in input character buffer. Reader is disconnected from buffer in off-line mode, and connected to output character buffer. Thus, no outputs can be produced to output character buffer while tape reader is operating off-line. Normally, off-line mode of tape reader is used with typewriter and type punch in off-line mode as shown in Table 8. Off-line reader operation is always pulsed mode, the typewriter or punch providing pulses for each character read.

2) TYPE

a) ON LINE/OFF LINE. Controls on- or off-line operation of input-output typewriter. For on-line mode operations characters are entered in input character buffer. Outputs from computer to typewriter are handled through the output buffer. During off-line operation, typewriter inputs and outputs are connected to the ouput buffer channel, and the keyboard lock is released.

3) PUNCH

a) ON LINE/OFF LINE. Controls on-line or off-line paper tape punch operation; in both modes punch is connected to standard character buffer; in off-line mode it can punch data from typewriter or tape reader as shown in Table.

b) ON/OFF. Controls operation of punch motor.

c) TAPE LOW. Shows when level of paper tape in punch has reached a pre-determined low level.

d) TAPE FEED. Momentary pushbutton which permits tape advance while only sprocket holes are punched.

MAINTENANCE PANEL

For computer maintenance and trouble shooting a separate maintenance panel is provided. See Figure 7. It is installed behind the second bay top panel of the computer cabinet. This panel swings into full view and access for computer diagnostics. The panel indicators and controls permit rapid localization of system failures.

Indicators

Neon indicator lights are provided which show the following:

1) MIR. Information register status.
2) MAR. Memory address register status.
3) AU Bus. Signals from AU bus.
4) TRANSFER BUS. Signals from main transfer bus.
5) READY/ENABLE I/O CHANNELS. Status of read and enable flip-flops of up to 8 input-output channels. Channel 1 is parallel channel in, channel 2 character buffer in, channel 3 parallel channel out, channel 4 character buffer out.
6) SHIFT COUNTER. Shows status of the 6 shift counter flip-flops.
7) CONTROL CLOCK. Shows fetch, transition and operate clock signals.

Controls

The following switches are provided:

1) INTERRUPT. On/off. Disable automatic interrupt capability of all input-output channels.
2) EXECUTE. Executes the op-code in the op-code register; when an operand address exists it will correspond to the 14 least significant bits, or address portion, of the Z-register if the index bit is ZERO. If the index bit is ONE the effective operand address is the sum of the address portion of Z and index register.
3) FETCH STOP. Stops command execution when a new fetch cycle is reached.

PAPER TAPE FORMAT

Paper tape coding is indicated in Figure 8. Punched holes in any of the eight paper tape channels correspond to ONE bits. Paper tape channel No. 1 corresponds to the least significant character bit.

Paper tape must be loaded with No. 1 next to the computer front panel.

When reading paper tape in octal format only the bits in channels 1, 2, 3 are read. During paper tape fill under control of the FILL button on the control panel the bits in channels 4, 6, 7, 8 must be ZERO. If any of these bits are ONE the associated character is ignored.

When reading paper tape in binary format the bits in channels 1, 2, 3, 4, 5, 6, 7 are read. Any character with a ONE in all of these positions is ignored and corresponds to the Delete character 011(p)1.111.

The parity bit is automatically checked while reading or generated while punching by the parity logic in the character buffer or word buffer. Odd parity is used.

If channel 8 contains a ONE bit (stop code), the paper tape reader is stopped. The associated character is not read by the computer.

For off-line paper tape preparation with the punch unit under direct keyboard control, the BACKSPACE key is to be used for a stop code. The typewriter code for BACKSPACE is 101100, but automatically an ad-

ditional ONE bit will be punched in paper tape channel 8. The stop code on tape will appear as: 110 (p) 1.100, with (p) corresponding to the parity bit. For on-line tape punching a stop code can be punched after setting the stop code flip-flop with an OCP command.

On-line typewriter input with the BACKSPACE key will also be interpreted as a stop code and will not be read by the computer. The occurrence of a stop code from on-line keyboard or paper tape reader can be tested with an SKS command.

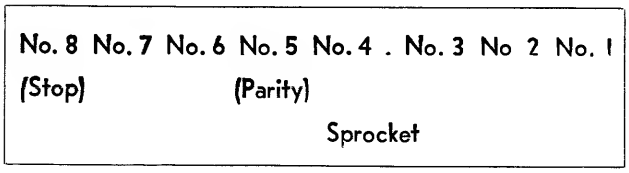


Figure 9 — Paper Tape Format

POWER FAILURE PROTECTION

The DDP-24 preserves the integrity of computation in event of power failure. The input power is monitored and if it should fall below the specified threshold an interrupt is executed. The interrupt routine stores the contents of the arithmetic and control unit and executes a HALT (HLT) instruction. When power rises above the threshold an interrupt may be initiated and an interrupt routine will restore the registers and resume operation.

OPERATION PROCEDURES

Power. Press the control panel POWER switch; then the MASTER CLEAR button. A running time indicator and the system circuit breaker are installed at the rear of the computer cabinet.

Paper Tape Program Loading

1) Load paper tape (always with three-channel side next to computer front panel).
2) Depress the program counter RESET button.
3) Manually enter into the program counter the memory address of the first paper-tape word to be stored.
4) Set paper tape reader control to ON-LINE and CONTINuous mode, then turn on tape reader power.
5) Press the FILL button to start tape moving through the reader.

   NOTE: Do not use the READER START button with this procedure.

6) The reader will stop automatically when it reaches the stop code which contains a ONE bit in the eighth paper tape channel.

During paper tape fill the HALT flip-flop will remain set. The characters are read from the tape, parity is checked, the characters are assembled into 24-bit computer words in the A-register and the words are entered in successive memory locations.

During fill, only the three least significant bits of each character are read (from the three tape channels on one side of the sprocket channel). These bits correspond to octal format; eight octal formats form a 24-bit word. Four of the five bits on the other side of the sprocket channel must be ZERO for the octal format; if any of these bits become ONE the buffer ignores the entire associated character.

TYPEWRITER PROGRAM LOADING

1) Press the MASTER CLEAR button and set the op-code register according to the OCP command.
2) Enter 00002000 into the Z-register (this corresponds to 02000 operand in address portion, for the OCP command).
3) Press the EXECUTE button on the maintenance panel (this seects typewriter input by executing instruction 05302000).
4) Press the program counter RESET button and manually enter the memory address for the first word from typewriter into the program counter.
5) Turn tape reader off.
6) Press the FILL button on the control panel.
7) Type in characters in octal from typewriter. Codes other than octal codes will be ignored. Eight characters are assembled into 24 bit computer words, starting from the most significant end of the word. The assembled words are entered in successive memory locations under automatic control of the FILL system.
8) The FILL mode will stop automatically when the BACKSPACE key (stop code) is depressed.

Manual Program Loading

1) Press the MASTER CLEAR button and enter the address information into the address portion of the Z-register.
2) Set the A-register according to the word to be entered in the computer.
3) Set the op-code register according to the STA command (code 05) and press the maintenance panel EXECUTE button.

   NOTE: This procedure loads only one word at a time; the effective address must be updated manually for consecutive words.

OPTIONAL PERIPHERAL EQUIPMENT

PAPER TAPE SPOOLER

A high speed paper-tape spooler with operating speeds up to 400 cps can be supplied with the DDP-24. The tape spooler handles 500 feet of 5 to 8 level tape in either direction. Fast rewind is provided at 1,000 cps.

DIGITAL X-Y PLOTTER

A digital plotter may be connected to the computer through the eight-bit character buffer. The solid-state plotter records 100 incremental steps per inch, at the rate of 200 per second, along one or both of two perpendicular axes in either positive or negative directions; Pen up and down positions may change at a rate of 10 operations per second; limit switches control maximum excursion. Normal chart width is 12 inches; a 31-inch width is available.

The plotter will plot points, continuous curves, curve identification symbols, letters, and numerals as directed by the DDP program.

MAGNETIC TAPE UNITS

Depending on the overall input-output rate required and the amount of computer time needed for further data processing and other computations, magnetic tape units of two types may be provided.

1) A transport operating at 30 inches per second, for both reading and writing with 200 and 555 bpi (bit-per-inch) tape density. Data transfer is 6,000 characters per second for 200 bpi tape density or 16,650 characters per second for 555 bpi.
2) A transport operating at 75 ips for both reading and writing with a 200 and 555 bpi tape density. The transfer data is 15,000 characters per second for 200 bpi tape density and 41,600 characters per second for 555 bpi.

Both magnetic tape units produce fully IBM 729 II compatible recording with either 200 or 555 bpi character density on $\frac{1}{2}$-inch, seven track magnetic tape; longitudal and character parity is provided. BCD character format with even parity and binary character format with odd parity is provided. Dual stack read-write heads allow for a parity read-after-write check.

The tape handlers have vacuum column tape buffer storage and solid-state electronics. Each unit may be connected to the DDP through a character input-output buffer, a word buffer, a Direct Memory Access Unit.

A properly coded OCP command starts the tape handler and permits characters to be read into the computer. Character parity is checked by the magnetic tape control unit; longitudinal parity at the end of the record is also checked. The longitudinal parity gap also stops the tape handler after each record is read. Although the tape can be read in either direction, the characters will be in reverse order if reading takes place in reverse; a one-character identifier, such as the file character, can of course be detected in either direction.

An OCP command initiates the write procedure after starting the tape handler in the forward direction; each character is written under control of a write

clock, which is part of the magnetic tape input-output control. The clock repetition rate can be selected to conform with the tape speed and recording density. A character parity bit is generated and written onto the tape with each character. The longitudinal parity character will be written automatically with the proper gap after the computer has signalled the tape unit that the recording has ended.

Under control of the DDP program the record gap signal and the file gap signal may be used to cause a seperate interrupt and is available to search for a given record or file. The computer program may keep track of the records being processed and search for records can easily be performed by counting the required number of gaps from the known position on tape. Backspacing of one record is handled with OCP commands. The tape unit will automatically be stopped by the end-of-tape signal. This signal too will cause an interrupt of the computer program. The record gap, file gap and end-of-tape signal will cause an interrupt with the same interrupt destination as the interrupt of the channel to which the tape unit is connected. Appendix C lists a number of OCP codes for magnetic tape unit control pulses.

DIGITAL RESOLVER

The 3C Digital Resolver is a special purpose digital device designed to generate a large number of algebraic, transcendental, and other mathematical functions. The Digital Resolver (DR) is a powerful satellite system for general purpose computers. It was designed specifically for the DDP-24 computer.

The DR can generate special functions while the computer is performing other tasks, and will perform these functions much faster than a general purpose computer. By the addition of a DR as a satellite to the DDP-24, the total increased speed in a real time problem solution varies from a factor of 2.5 to 10 in typical cases.

Examples of functions particularly suited for the DR are trigonometric, hyperbolic, and logarithmic functions, exponentials, Cartesian to polar, Cartesian to rotated Cartesian, and polar to Cartesian coordinate conversions. Typical total solution times for these tasks are between 180 and 470 μsec.

Additional information on the Digital Resolver is available through other 3C publications.

At the beginning of each DR operation the Digital Resolver must be properly loaded. There are three main registers which must be loaded with information from the DDP-24, reset, or left unchanged. The DR contains a serial-parallel buffer register providing interface with the DDP-24 for loading and storing the results of a DR operation. The DR is connected to the DDP-24 with one parallel input channel and one parallel output channel via the serial-parallel buffer register.

Depending on the number of registers to be loaded, the load subroutines take an average of 30 μsecs.

MISCELLANEOUS EQUIPMENT

Because of the flexible input-output capabilities of the DDP-24 an almost unlimited range of peripheral devices may be connected to it. Typical examples are card readers, punches, high speed line printers, additional I/O typewriters, analog-to-digital and digital-to-analog converters. A DDP-24 computer may also be operated as an input-output device for another DDP-24, forming a master-slave combination.

PROGRAMMING

INTRODUCTION

A comprehensive package of software programs has been written for the DDP-24 and is available to the user as part of the standard system. These programs have been designed for a wide range of user skills; the most widely used programming practices and conventions have been incorporated. The FORTRAN II compiler has been augmented with FORTRAN IV conventions for logical and Boolean operations. The compiler requires 4,000 words of memory and will operate on the basic paper tape, typewriter system. The assembly program provides one-to-one translation of symbolic statements plus programmer defined macro capabilities; linkage compatibility is provided so that programs written in the assembly language can be called by FORTRAN programs. The output is either absolute or relocatable at the programmer's option. The assembler requires 1,500 words of memory plus symbol tables which vary in size depending on the amount of memory available. The interpretive program requires 1,000 words of memory and permits the user with minimum programming experience to easily learn and use the computer for scientific computations. The executive control program requires 300 words of memory and permits the flexibility of on-line communication with the computer during operation. Library subroutines, utility programs, input-output routines and diagnostic routines are also provided. All of the programming systems described in this manual will operate with the standard DDP-24. They will utilize additional core memory and magnetic tapes when available.

FORTRAN II COMPILER

The DDP-24 FORTRAN II language is compatible with the generally accepted standards of FORTRAN II compiler used on large-scale computers except for double precision and complex arithmetic. Additional features have been added which include the following:

1) Type (Declarative) Statements

   These statements follow the FORTRAN IV conventions of LOGICAL, REAL, and INTEGER. Thus, logical variables may be defined and the convention of beginning integer variables with I, J, K, L, M or N can be overridden by the use of the REAL and INTEGER type statements.

2) Logical Expressions

   A logical expression is one that uses the following operators:

   .AND. — used for forming logical products

   .OR. — used for forming logical sums

   .NOT. — used for forming logical exclusive sums

   .SHFT. — used for shifting the bit pattern in any variable right or left (specified by the sign of the expression to the right of the .SHFT. operator).

3) Mixed Expressions

   Expressions involving the combination of integer, real or logical variables may be used. For example, if K is a logical variable, M is an an integer variable and A is a real variable, the expression A=K*M will cause A to be replaced (in floating point) by either O or M depending

expression A=K*M will cause A to be replaced (in floating point) by either O or M depending on the value of K. Thus, it can be seen that Boolean operations are offered as a subset of logical operations.

4) Full Word Integers

The DDP-24 FORTRAN II compiler will produce object programs that utilize full word integer arithmetic (sign plus 23 bits).

5) Compatibility

The language is such that if attention is paid to the selection of statements, the source program can be compiled by the 7090 FORTRAN II commpiler. Also, FORTRAN II programs written for other computers can be compiled on the DDP-24 if the standard FORTRAN II conventions are followed.

6) Machine Language In-Line with FORTRAN Statements.

In-line machine language statements will be translated by the DDP-24 FORTRAN II compiler. This feature is extremely important in many real-time applications.

7) Input and Output

Input of the source language can be either paper tape, cards or magnetic tape. Symbolic output will be either a typewriter listing of the source program (programmer's option) or a magnetic tape for off-line listing. Output of the relocatable object program can either be paper tape, cards or magnetic tape.

8) Assembly Language Output

The compiler can produce a symbolic language output rather than the object program. This symbolic output can later be assembled by the assembly program. Thus, it will be possible to modify FORTRAN produced programs using assembly language.

9) Multiple Program Compilations

Multiple program compilation without operator intervention is provided when magnetic tape is used for the source language input.

10) Load and Go

The stored programs may be loaded, compiled, and executed without operator intervention when magnetic tape is used for both input and output (two tape units).

DDP-24 ASSEMBLY PROGRAM (DAP)

DAP will assemble a symbolic language that permits the writing of a machine language program in a form that is more convenient to the programmer than numeric coding. It allows for the substitution of meaningful mnemonic symbols for desired binary instructions, (e.g. ADD in place of 001000) and allows the programmer to assign names to specific data items or groups such as DAY, RATE, PI, and to use these names when referring to the items as operands. However, since DAP is closely related to the basic machine language it in no way inhibits the programmer's use of the computer hardware facilities. DAP is designed to work with the minimum machine configuration: paper-tape reader, paper-tape punch, on-line typewriter, and 4,096 words of memory. No optional features are required to assemble a program using DAP, however, magnetic tapes and card equipment may be used if available. Thus the user may assemble a program designed for a more complex system on a minimum capacity computer. In addition to the one-to-one translation of machine language mnemonics, DAP also translates programmer defined macro-operations and the following pseudo-operations:

Pseudo-Operation	Function
ABS	Produce an absolute object program
BCI	Define alphanumeric data
BES	Define a block of storage and label the ending cell
BSS	Define a block of storage and label the starting cell
CALL	Produce the linkage necessary for transferring control to a library subroutine and include the library subroutine as part of the object output
COMN	Define a common data area that can be referenced by other programs
DEC	Define decimal data
END	End of source program
ENM	End of macro definition
EQU	Assign a fixed value to a variable symbol
LIST	Produce a program listing
MAC	Beginning of a macro definition
MOR	Stop reading the source tape until the START button on the operator's console is depressed
MZE	Define a MINUS ZERO
NLST	Do not produce a program listing
NTRY	Define the start of a library subroutine
OCT	Define octal data

ORG	Define a starting memory location for the instructions that follow
PZE	Define a plus zero
REL	Produce a relocatable object program

Additional features of DAP include the following:

1) Literals

Literal constants are allowed as elements of the address field. Literals are preceded by the equals character (=) and may be either decimal, octal or alphanumeric.

2) Complex Address Fields

Algebraic expressions may be used in the address field. The following operators allowed:

* multiplication
/ division
\+ addition
— subtraction

3) SHARE Conventions involving the asterisk (*)
The following SHARE conventions using the asterisk are allowed by DAP:

a) * in column 1 or first element in location field — treat entire card or line as comment

b) * appended to instruction mnemonic — set indirect address flag

c) * as first element in address field — means "this location"

d) ** as a symbolic address — address will be modified by another instruction

e) *** as an operation code — op-code will be modified by another instruction

4) FORTRAN Compatible

The CALL and NTRY pseudo-operations produce FORTRAN compatible linkages so that machine language or FORTRAN subroutines can be used interchangeably.

5) Input and Output

Input of the source language can either be paper tape, cards or magnetic tape. Symbolic output can either be a side-by-side listing of the source and object program (programmer's option) or a magnetic tape for off-line listing. Output of the relocatable or absolute object program can either be paper tape, cards or magnetic tape.

6) Multiple Program Assemblies

Multiple program assemblies without operator intervention is provided when magnetic tape is used for the source language input.

7) Load and Go

The assembled object programs may be loaded, assembled, and executed without operator intervention when magnetic tape is used for both input and output (two tape units).

DDP-24 INTERPRETIVE PROGRAM (DIP)

This programming system is especially easy to use; it serves both as an educational tool and as an introduction to the fundamentals of an internally programmed digital computer. The customary extensive programming rules, number system conversions, and scaling, troublesome to the beginner or occasional user, are integral to the system and need not be of concern. DIP operates on the standard DDP-24 and expanded systems. Some features are:

1) Macro command format
2) Decimal input and output
3) Typewriter and paper tape input-output
4) Logical operations under program control
5) Double precision floating point arithmetic and mathematical subroutines available in DAP and FORTRAN II
6) Index register operation with 10 program index registers
7) Program debugging and trace features including breakpoint halt and capability to examine instructions or registers on request
8) Provision for restarting computation in event of error
9) Provision to utilize extended memory
10) Provision for DIP library extension

EXECUTIVE CONTROL PROGRAM (DEP)

This self-loading program assists the programmer in his communication with the computer during actual operation. It interprets certain commands entered through the on-line typewriter and automatically performs the desired operations. Upon transferring control to DEP, the following will be typed on the typewriter:

FUNCTION

The programmer then types the name of the function desired. DEP has a modular structure which can be easily expanded. A brief description of the functions included as part of the basic DEP dictionary follows:

Dump

In response to this function request, the typewriter will ask for LIMITS. The programmer then enters the starting and stopping inclusive limits of memory that are to be dumped. DEP will perform the desired

dumping operation on the typewriter in octal, eight words per line. Upon reaching the terminal limit, a new function will be requested.

Load

In response to this function request, the typewriter will ask for MODE. The programmer then enters either RELOCATABLE or ABSOLUTE depending on the tape mode to be loaded. If the programmer types ABSOLUTE, DEP will immediately start loading the object tape into memory at the location absolutely specified on the tape; if the programmer types RELOCATABLE, the typewriter will request STARTING LOCATIONS. The programmer will type the starting location and DEP will load the relocatable tape into the memory starting at that location. Upon reaching the end of the tape, a new function will be requested.

Duplicate

In response to this function request, DEP will immediately start reading the tape in the paper tape reader and punch a duplicate copy on the paper tape punch. Upon reaching the end of the tape, a new function will be requested.

Punch

In response to this function request, the typewriter will ask for LIMITS. The programmer then enters the starting and stopping inclusive limits of memory that are to be punched. DEP will perform the desired punching operation on the paper tape punch, producing an absolute object tape that may be loaded at some future time by using the LOAD function. Upon reaching the terminal limit, a new function will be requested.

List

In response to this function request, the typewriter will ask for FORMAT. The programmer then types BCD, OCTAL or BI-OCTAL. These specifications have the following meanings:

BCD—Read the tape in the paper tape reader, 6 bits per frame, and type the corresponding alphanumeric characters. Control codes such as carriage return and tab will not be typed, but will cause these functions to be performed on the typewriter.

OCTAL—Read the tape in the paper tape reader, 3 bits (low order) per frame, and type the corresponding octal digits, eight digits (one word) per line.

BI-OCTAL—Read the tape in the paper tape reader, 6 bits per frame, and type the corresponding two octal digits, eight digits (one word) per line. Upon reaching the end of the tape, a new function will be requested.

JUMP TO nnnnn—In response to this function request, DEP will transfer control to the instruction in the memory location specified by nnnnn (in octal). Execution will proceed from that point and DEP will no longer have control of the computer.

LIBRARY SUBROUTINES AND UTILITY PROGRAMS

Mathematical

The following mathematical routines are available in fixed-point single and double precision, and floating-point single and double precision:

1) Log (natural and common)
2) Exponential
3) Sine (radians)
4) Cosine (radians)
5) Tangent (radians)
6) Tangent $^{-1}$ (radians)
7) Square root
8) Add, Subtract, Multiply and Divide (not necessary for fixed-point single precision).

Conversion

1) BCD to single or double precision floating-point
2) BCD to single or double precision fixed-point
3) Single or double precision floating-point to BCD
4) Single or double precision fixed-point to BCD
5) single or double precision fixed-point to single or double precision floating point
6) single or double precision floating-point to single or double precision fixed-point.

Input-output

1) Typewriter in or out
2) Paper tape in or out
3) Magnetic tape in or out
4) BCD field insertion for formatting output

Utility

1) Memory dump that produces a listing in octal, decimal (single or double precision, fixed or floating), alphanumeric, instruction mnemonics.
2) Loader for absolute or relocatable programs
3) Library updating and editing
4) Source program updating and editing

Test and Maintenance

1) Core memory check routine
2) Central processing unit check routine
3) Typewriter check routine
4) Paper tape reader check routine
5) Paper tape punch check routine
6) Magnetic tape check routine

APPENDIX A

COMPUTER CODES

Digital computers in most cases use a binary number code. In this code, each binary digit or bit in a number corresponds to a power of 2, with the least significant bit (at the right) corresponding to 2^0 or 1. Only the digits 1 and 0 are used.

Depending on how negative numbers are represented, there are different binary codes possible, of which the most important ones are the two's complement, one's complement and sign magnitude codes.

In the two's complement code each number, positive or negative, always corresponds to its next higher binary number decremented with 1, or its next lower binary number incremented with 1, neglecting any overflow. This is not changed by transition from positive to negative value and conversely.

The one's complement code provides for a simpler representation of negative numbers. A negative number corresponds to the same positive number, except that all bits are complemented, for example, ONES become ZEROS and vice versa. The sign bit naturally is also complemented. This code requires a correction in the normal binary additions and subtractions by the computer whenever a transition from a positive to a negative value takes place.

The simplest code for the human operator or programmer is the sign magnitude code. The magnitude portion is the same for either positive or negative numbers. Only the sign bit differs (1 for negative numbers, 0 for positive numbers). This code comes closest to the everyday manner of representing positive and negative numbers. In terms of computer logic it requires the same correction for transitions from positive to negative numbers and vice-versa plus complementing of negative numbers.

Because it is impractical in a serial computer to apply any corrections as in one's complement or sign magnitude codes, these computers generally use the two's complement code.

Parallel machines most often use the one's complement code. The DDP-24 is the only machine in its general class using the convenient sign magnitude code.

APPENDIX B

COMMAND LIST

In numerical order the DDP command list is as follows:

Op	Command	Description	Execution Time
00	HLT	Halt	5 μsec
02	XEC	Execute	5 μsec + variable
03	STB	Store B	10 μsec
04	STC	Store Op-Code Portion of A	10 μsec
05	STA	Store Address Portion of A	10 μsec
06	STD	Input to Memory	10 μsec
07	INM	Store A	10 μsec
10	ADD	Add	10 μsec
11	SUB	Subtract	10 μsec
12	SKG	Skip if A Greater	10 or 12 μsec
13	SKN	Skip if A not equal	10 or 12 μsec
15	ANA	AND to A	10 μsec
16	ORA	OR to A	10 μsec
17	ERA	Exclusive OR to A	10 μsec
20	ADM	Add Magnitude	10 μsec
21	SBM	Subtract Magnitude	10 μsec
22	OTM	Output from Memory	10 μsec
23	LDB	Load B	10 μsec
24	LDA	Load A	10 μsec
25	JRT	Jump Return	10 μsec
27	JST	Jump and Store Location	10 μsec
30	SMP	Step Multiple Precision	10 μsec
31	FMB	Fill Memory Block	variable
32	DMB	Dump Memory Block	variable
34	MPY	Multiply	31 μsec
35	DIV	Divide	33 μsec
36	BCD*	BCD to Binary Conversion	33 μsec
37	BIN*	Binary to BCD Conversion	33 μsec
40	ARS	A Right Shift	5+n μsec
41	ALS	A Left Shift	5+n μsec
42	LRR	Long Right Rotate	5+n μsec
43	LLR	Long Left Rotate	5+n μsec
44	LRS	Long Right Shift	5+n μsec
45	LLS	Long Left Shift	5+n μsec
46	NRM	Normalize	variable
47	LGL	Logical Left Shift	5+n μsec
50	OTA	Output from A	5 μsec
51	ITC	Interrupt Control	5 μsec
52	INA	Input to A	5 μsec
53	OCP	Output Control Pulse	5 μsec
54	ADX	Add to Index	5 μsec
55	TAB	Transfer A to B	5 μsec
56	LDX	Load Index	5 μsec
57	IAB	Interchange A and B	10 μsec
60	CRA	Clear A	5 μsec
61	SKS	Skip if Sense Line Not Set	6 μsec
62	RND	Round A	6 μsec
63	TAX	Transfer A to Index	5 μsec
64	SCR	Scale Right	5+n μsec
65	SCL	Scale Left	5+n μsec
66	STX	Store Index	10 μsec
67	IRX	Increment, Replace and Load Index	14 μsec
70	JPL	Jump if A Plus	6 μsec
71	JZE	Jump if A Zero	5 μsec
72	JIX	Jump on Index	5 μsec
73	JOF	Jump on Overflow	5 μsec
73	JOF	Jump on Overflow	5 μsec
74	JMP	Unconditional Jump	5 μsec
75	JXI	Jump on Index Incremented	7 μsec
77	NOP	No Operation	5 μsec

* Optional Commands

APPENDIX C

OCP ADDRESS CODES (x=unit number)

The following is a list of assigned codes to be used in the address portion of the OCP instruction to perform the specified functions. The codes are given in octal notation; x indicates number of unit.

BOTH INPUT AND OUTPUT CHANNELS

00000	Enable Standard I/O Character Channels
00001	Enable #1 Optional I/O Character Channels
thru	thru
00007	Enable #7 Optional I/O Character Channels
00010	Enable Standard I/O Word Channels
00011	Enable #1 Option I/O Word Channels
thru	thru
00076	Enable #54 Optional I/O Word Channels
00077	Inhibit all I/O Channels

INPUT CHANNELS ONLY

00100	Enable Standard Input Character Channel
00101	Enable #1 Optional Input Character Channel
thru	thru
00107	Enable #7 Optional Input Character Channel
00110	Enable Standard Input Word Channel
00111	Enable #1 Optional Input Word Channel
thru	thru
00176	Enable #54 Optional Input Word Channel
00177	Inhibit all Input Channels

OUTPUT CHANNELS ONLY

00200	Enable Standard Output Character Channel
00201	Enable #1 Optional Output Character Channel
thru	thru
00207	Enable #7 Optional Output Character Channel
00210	Enable Standard Output Word Channel
00211	Enable #1 Optional Output Word Channel
thru	thru
00276	Enable #54 Optional Output Word Channel
00277	Inhibit all Output Channels

WORD BUFFER

003mx	Word Buffer Control (8 possible) m=number of characters per word

MISCELLANEOUS

01000	Enable Stop Code Punch
01001	
thru	General Purpose Control Pulses for External Devices
01777	01001 thru 01007 Standard with DDP-24

TYPEWRITER

0200x	Typewriter Input Select (Keyboard Enabled)
0201x	Typewriter Output Select (Keyboard Inhibited)
0207x	Typewriter Disconnect (Keyboard Released)

NOTE: x=0 corresponds to standard typewriter

PAPER TAPE READER

0210x Paper Tape Reader Start
0217x Paper Tape Reader Stop
NOTE: x=0 corresponds to standard paper tape reader

PAPER TAPE PUNCH

0220x Paper Tape punch Select
NOTE: x=0 corresponds to standard paper tape punch

LINE PRINTER

0230x thru 0237x Card Reader Control

CARD READER

0240x thru 0247x Card Reader Control

CARD PUNCH

0250x thru 0257x Card Punch Control

DIGITAL X—Y PLOTTER CONTROL (2 possible)

026nn Plotter #1
027nn Plotter #2
nn Specifies Function as Follows:
01 Step —Y (carriage right)
02 Step + Y (carriage left)
04 Step — X (drum up)
05 Step —X, —Y
06 Step —X, +Y
10 Step +X (drum down)
11 Step +X, —Y
12 Step +X, +Y
20 Plotter pen down
40 Plotter pen up

MAGNETIC TAPE CONTROL

03x00 thru 03x17 Subselect one of up to 16 tape handlers connected to the same control unit; X specifies the control unit; 8 magnetic tape control units are possible
03x23 Start tape handler and read one block, even parity (to be preceded by OCP instruction with either 03x42, 03,43, 03x52 or 03x53)
03x24 Start tape handler and write one block, even parity (to be preceded by OCP instruction with either 03x42 or 03x52)
03x25 Start tape handler and read one block, odd parity (to be preceded by OCP instruction with either 03x42, 03x43, 03x52 or 03x53 in the address portion)
03x26 Start tape handler and write one block, odd parity (to be preceded by OCP instruction with either 03x42 or 03x52 in address portion)
03x41 Move one block (to be preceded by OCP instruction with either 03x42, 03x43, 03x52 or 03x53)
03x42 Forward with interrupt by record gaps
03x43 Reverse with interrupt by record gaps
03x44 Search for file gap
03x50 Stop transport
03x51 Start transport (to be preceded by either 03x42, 03x43, 03x52 or 03x53)
03x52 Forward
03x53 Reverse
03x54 Fast Forward
03x55 Fast Reverse
03x56 Rewind to load point
03x61 Reset write flip-flop

A/D and D/A CONTROL

04000 thru 04377 A/D Control (to be specified)
04400 thru 04777 D/A Control (to be specified)

DIGITAL RESOLVER

05000 T mode with prescaling
05001 T mode without prescaling
05002 T* mode with prescaling
05003 T* mode without prescaling
05004 H mode with prescaling
05005 H mode without prescaling
05006 H* mode with prescaling
05007 H* mode without prescaling
05010 Sequential load of DR; start with Y
05011 Sequential load of DR; start with X
05012 Sequential load of DR; start with W
05013 Sequential DR output; start with Y
05014 Sequential DR output; start with X
05015 Sequential DR output; start with W

CONTROL OF DIRECT MEMORY ACCESS AND FULLY BUFFERED CHANNELS

0600n thru 0603n Direct Memory Access Channels
0604n thru 0607n Fully Buffered Channels
n=function specification

APPENDIX D

SENSE LINE CODES

The 14-bit address portion of the SKS command represents sense lines as follows:
Internal Sense Lines (simultaneous tests possible):

Sense Switches

0	0	unused	unused	stop code	over flow	improper divide	parity error	#6	#5	#4	#3	#2	#1

Channel Ready Signals (simultaneous tests possible):

ready signals of additional I/O channels

0	1	Par. Chan. In	Char. Buff. In	Par. Chann. Out	Char. Buff Out						

External Sense Lines

The address portion of the SKS command specifies 16 sense lines as follows (octal code):

$2\,x_1\,x_2\,x_3\;0$ thru $2\,x_1\,x_2\,x_3\;7$

$3\,x_1\,x_2\,x_3\;0$ thru $3\,x_1\,x_2\,x_3\;7$

The x_1, x_2, and x_3 octal digits corresponding to bits 13-21 of the DDP word are not normally used in the testing, but each of the 16 sense lines may actually represent a group of 512 different test signals when x_1, x_2, x_3 are specified. For this purpose bits 13-21 are available for external matching with test signals.

NOTE

If the index bit of the SKS command word is ONE, any flip-flop tested by the SKS command is also reset by it.

APPENDIX E

DDP-24 CHARACTER CODES

OCTAL CODE	TYPEWRITER L/C	TYPEWRITER U/C	PAPER TAPE 8	7	6	P	4	S	3	2	1	
00	Ø	b				O		•				
01	1							•			O	
02	2							•		O		
03	3					O		•		O	O	
04	4	:						•	O			
05	5	@				O		•	O		O	
06	6	√				O		•	O	O		
07	7	>						•	O	O	O	
10	8						O	•				
11	9					O	O	•			O	
13	#	-					O	•		O	O	
20	*	¢			O			•				
21	/				O	O		•			O	
22	S				O	O		•		O		
23	T				O			•		O	O	
24	U	=			O	O		•	O			
25	V	%			O			•	O		O	
26	W	"			O			•	O	O		
27	X	'			O	O		•	O	O	O	
30	Y				O	O	O	•				
31	Z				O		O	•			O	
33	'				O	O	O	•		O	O	
40	-	-		O				•				
41	J			O		O		•			O	
42	K			O		O		•		O		
43	L			O				•		O	O	
44	M	)		O		O		•	O			
45	N	*		O				•	O		O	
46	O	△		O				•	O	O		
47	P	;		O		O		•	O	O	O	
50	Q			O		O	O	•				
51	R			O			O	•			O	
52	tab			O			O	•		O		
53	$			O		O	O	•		O	O	
54	backspace*			O			O	•	O			**
56	space			O		O	O	•	O	O		
60	&	&		O	O	O		•				
61	A			O	O			•			O	
62	B			O	O			•		O		
63	C			O	O	O		•		O	O	
64	D	(		O	O			•	O			
65	E	□		O	O	O		•	O		O	
66	F	/		O	O	O		•	O	O		
67	G	<		O	O			•	O	O	O	
70	H			O	O		O	•				
71	I			O	O	O	O	•			O	
73	.	∨		O	O		O	•		O	O	
74	lower shift			O	O	O	O	•	O			
75	upper shift			O	O		O	•	O		O	
76	car. return			O	O		O	•	O	O		
77	line feed			O	O	O	O	•	O	O	O	
stop	backspace*		O	O		O	O	•	O			